2018 Sagan Summer Workshop -- EXOFASTv2 Hands on Session

Jason Eastman, Joseph Rodriguez

During the EXOFASTv2 portion of the hands-on session, we will show how to model RVs, transits, and the host star using EXOFASTv2, a sophisticated global modeling package written in IDL using a differential evolution MCMC method to determine the median and 68% confidence intervals of each parameter.

HAT-P-3b Walkthrough

We will spend ~30 minutes walking through this simplified fit of HAT-P-3b and discuss the minimum requirements for performing a fit and basic quality control methods to make sure the fit makes sense. To start the fit, type:

```
cd $EXOFAST_PATH/examples/hat3
idl
fithat3
```

Wait for it to finish, then inspect most relevant outputs:

```
cd $EXOFAST_PATH/examples/hat3/fitresults/
okular HAT-3b.Torres.mcmc.rv.ps &
okular HAT-3b.Torres.mcmc.transit.ps &
okular HAT-3b.Torres.chain.ps &
okular HAT-3b.Torres.pdf.ps &
okular HAT-3b.Torres.covar.ps &
emacs HAT-3b.Torres.median.tex &
```

Compile latex table of outputs and display

```
pdflatex HAT-3b.Torres.median.tex
okular HAT-3b.Torres.median.pdf &
```

Goal: Develop a baseline understanding of exoplanet modeling, EXOFASTv2, MCMC modeling, and quality control, including chains, links, burn-in, gelman-rubin statistic, number of independent draws, convergence, and thinning.

EXOFASTv2 is powerful, but complex. It’s easy to get into trouble if you don’t know what you’re doing. You should understand each parameter that has to be set and its effect on the fit. As such, we have created several different exercises with different goals and with different levels of experience in mind.

On your own

Each group will then select a system from $EXOFAST_PATH/examples/ and one of the following questions to address. These are ordered from easiest to hardest. Novice users should not attempt 7-9:

1. For stars that have transit or RV data from multiple telescopes (KELT-4,KELT-6,KELT-14,WASP-76), see how the fit differs if you use data from only one telescope, then from all telescopes together. What additional parameters are added for each RV instrument/telescope?
2. For stars with transits in multiple photometric bands (KELT-6, KELT-16), see how the fit differs if you fit one band at a time, then using the transit from all bands.
3. Run fits with different numbers of maximum steps for the MCMC fit. Compare the results; also take note of the difference between the chi-squared fit and the MCMC fit.
4. For transit data taken in long cadence (GJ9827, EPIC 247098361, HD106315), fit it with and without taking the long cadence smearing into account. What biases are introduced by ignoring the long exposure times?
5. Change the starting values for the fits and see how the fit is impacted, both in terms of speed and final results.
6. Perform different fits using the Torres relations, YY isochrones, and MIST isochrones to constrain the stellar parameters. Set nplanets=0 to fit just the star. How do the results compare? Which methods are faster? Why?
7. Add a flux file with the broad-band magnitudes of the host star and see whether your fit improves. In particular, how does the result change when you use a parallax constraint from Gaia DR2 and upper limit on the V-band extinction from Schlegel dust maps?
8. For stars that have transit and RV data (EPIC 247098361, HAT-3, HATS-7, KELT-1, KELT-14, KELT-16, KELT-6, WASP-76): run a fit using only the transit data, then a fit using only the RV data, and finally a fit with the transit and RV data. How do the results compare?
9. For multi-planet systems (KELT-6, HD106315, GJ9827), fit each planet individually, then fit them all together. How do the results compare?

ALL: How do your results compare with the literature values? The discovery paper is included in the examples directory for each system.

BONUS: EXOFASTv2 can be run without an IDL license (i.e., for free). Try any of the above, following the license-free example in $EXOFAST_PATH/examples/hat3_nolicense/fithat3.sh.

You will do the following:

- run the baseline (pre-prepared) fit

- while that’s running, tweak the pro file and priors file as necessary to achieve your goals. Be sure to change the “prefix” between fits so new fits do not overwrite old results. Detailed instructions to run a fit starts on the next page.

NOTE 1: Each baseline fit has been tuned to take ~25 minutes to complete without running over the 2GB memory limit, but that does not mean it is good enough for publication/use. The more complicated the system/model, the less likely it is to be well-mixed. Heed the warnings about the Gelman Rubin statistic and make sure you inspect the results! For some systems, due to the time constraints, it will likely be necessary to present unmixed results. Include that caveat in your presentation.

NOTE 2: If you’re doing well on time, try creating the pro file and prior file from scratch. Compare with the supplied files.

General steps for an EXOFASTv2 fit

The fit is run with a “fit_planet-name.pro” file. The data (RVs, transit light curves, and/or broad band photometry) are in text files, all of which are optional. The data filenames are expected to follow a certain format, explained below. There is another text file with priors that specifies the starting values and external constraints for the fit and is required.

By default, the output files are put in a subdirectory, “fitresults” which is created if it doesn’t already exist, and can be changed with the “prefix” input set in the .pro file. Initially, several files with the prefix and "start" prepended to the filename are created, which show the starting model over the input data.

Then, after the amoeba fit finishes (typically a few minutes), new versions of each file with the same prefix and "amoeba" are generated, showing the best-fit model found by amoeba overlaid on the data. Finally, after the MCMC is done (which can take tens of minutes or weeks), a third version of each file with the same prefix and “mcmc” are created, as well as several diagnostic files. The final file that is generated is the prefix.chain.ps file.

1. Prepare the transit data (optional)
    a. The transit data must be in as ascii plain text file with 3 columns.
        i. The first column must be BJD_TDB
            1. BJD_TDB takes into account the light travel time and relativity to ensure the observed times are not distorted by the moving Earth
            2. Other input time stamps will likely not break the code, but can introduce errors in the reported times of more than a minute. The output table will display BJD_TDB but makes no attempt to convert input times.
            3. For more information, see:
                a. https://arxiv.org/abs/1005.4415
            4. BJD_TDB can be calculated from JD_UTC here:
                a. http://astroutils.astronomy.ohio-state.edu/time/utc2bjd.html
        ii. The second column must be flux, with the out of transit data normalized to 1
        iii. The third column must be flux uncertainty
            1. A variance will be fitted to scale the error, so an estimate is ok here.
            2. The standard deviation of the out of transit points is sufficient if you don’t have an independent estimate of the uncertainty.
    b. Each file is able to be fit with an independent wavelength, error scaling, transit time, transit depth, and/or transit duration, so you may include multiple transits in the same file as long as they’re in the same wavelength, have the same baseline flux, and you don’t want to fit TTVs. Otherwise, each transit should have its own file.

c. The name of the transit file *must* adhere to a certain format: nYYYYMMDD.filtername.telescope.whateveryouwant.
    i. nYYYYMMDD -- The UTC date of mid transit, where YYYY is the year, MM is the month, and DD is the day. This is used to label the transits in the output plot and tables
    ii. filtername -- The name of the observed filter. This is used to define the limb darkening for the transit. In addition, a separate dilution and secondary depth can be fit for each filtername. Only certain values are allowed (use the closest approximation if yours is not in this list see quadld.pro)
        1. Johnson/Cousins: 'U','B','V','R','I','J','H','K'
        2. Sloan: 'Sloanu','Sloang','Sloanr','Sloani','Sloanz'
        3. Kepler: 'Kepler’
        4. CoRoT: 'CoRoT’
        5. TESS: 'TESS'
        6. Spitzer: 'Spit36','Spit45','Spit58','Spit80'
        7. Stromgren: 'u','b','v','y'
    iii. telescope -- a description of the telescope used for the observations. Anything is allowed, but all observations observed with the same telescope should have the same name. This is used in the output plot and color codes the TTV plot.
    iv. whateveryouwant -- any string you want to include for it to make sense to you. This is not used by the code.
    v. Periods are delimiters that should not be used except as outlined above.
    vi. So a transit taken on UTC 2017-01-27 with MINERVA in the V band would be "n20170127.V.MINERVA.dat"

d. The units must be as specified, or the fit will be wrong or fail.

e. If no transit files are provided, no transit will be fit
    i. If RVs are provided, it will determine the mass, and estimate the planet radius and derived quantities based on the Chen & Kipping mass radius relation
        1. http://adsabs.harvard.edu/abs/2017ApJ...834...17C

f. In the call to EXOFASTv2, all fitted transits are specified in the “TRANPATH” using a single wildcard (e.g., tranpath=‘n*.dat’).

2. Prepare the RV data (optional)
    a. The RV data must be in as ascii plain text file with 3 columns
        i. The first column must be BJD_TDB
            1. See notes above
        ii. The second column must be RV, in m/s
        iii. The third column must be uncertainty, in m/s
            1. A jitter variance will be fit, so an estimate is fine
    b. The units must be as specified, or the fit will be wrong or fail.
    c. If no RV files are provided, no RV will be fit.

        i. If transits are provided, it will determine the radius, and estimate the planet mass and derived quantities based on the Chen & Kipping mass radius relation
            1. http://adsabs.harvard.edu/abs/2017ApJ...834...17C
    d. The name of the RV file should adhere to a certain format: planet_name.telescope.whateveryouwant
        i. "planet_name" is the name of the planet (e.g., HAT-3b). This is not used by the code, but is good practice to include here.
        ii. "telescope" is the name of the telescope or instrument used to make the observations. This will become the entry for the legend in the RV plot (if multiple instruments are supplied) and the column label in the output table.
        iii. whateveryouwant -- any string you want to include for it to make sense to you. This is not used by the code. I recommend using a common prefix (e.g., ".rv") so it can be specified cleanly in RVPATH (e.g, RVPATH='*.rv').
        iv. Periods are delimiters that should not be used in the filename except as outlined above.
        v. So TRES observations of HAT-3b might be called
            HAT-3b.TRES.rv
3. Prepare the prior file
    a. Priors and starting values for the fit are modified via the 'priorfile'.
    b. "#" is a comment character. Inline comments are allowed.
    c. You may have 2-5 white-space delimited entries per line
        i. Which parameters you must and should include here and how depends strongly on what data sets you're fitting and what your goals are. See the examples for good practices on similar data sets.
        ii. The first entry must match a parameter name used in the code
            1. Fitted or derived parameters can be used
                a. A full list of parameters is here:
                    i. https://github.com/jdeast/EXOFASTv2/blob/master/parnames.README
                b. Changing the starting value of derived parameters may not always have the intended effect
                    i. Most reasonable parameters one would typically change are supported
                    ii. But some are not. Sometimes these are just unsupported (probably an indication you shouldn't be doing that…), sometimes the number of ways to change the starting values based on derived parameters is too numerous to practically support

        iii. Check the output at the start of the fit to ensure it was properly applied. If not, change the starting values of the fitted parameters instead.
    2. Non-matching names will be ignored with a warning.
    3. Case is ignored
        a. mstar = MStar = MSTAR = mStAr
    4. When fitting datasets with multiple telescopes, planets, wavelengths, or transits, you must append a suffix, (e.g., “_1”) to the corresponding parameter name to specify which.
        a. The suffix is a zero-based index number. The appropriate indices corresponding to each object are generally alphabetical, but will be printed at the start of the fit

iii. The second entry is the starting value.
    1. This changes the default starting value of the fit, but (without a third entry) does not penalize the fit for straying from it.
    2. When a third value is also specified, this starting value doubles as the mean of the Gaussian prior.
    3. There is no option to ignore this entry.

iv. The third entry is a gaussian prior width.
    1. A penalty is added at each step equal to ((step_value - starting_value)/width)^2.
    2. Any negative value (e.g., -1) will be ignored, allowing the user to specify bounds (4th and 5th entries) without applying a gaussian prior.
    3. Asymmetric error bars are not supported.
    4. A prior width of 0 will fix the parameter to the starting value.
        a. Fixing parameters is generally not recommended because you will underestimate the uncertainties of any covariant parameter
        b. Applying a prior with a realistic uncertainty is strongly preferred. If possible, using the data set from which the prior was derived is even better.
    5. The most common error I see is applying a prior based on a similar analysis of the same data (e.g., imposing a period prior from K2 data when fitting the K2 data, imposing an SED-derived prior on Rstar while fitting the SED, or imposing a prior on the stellar mass/radius from an evolutionary track while using the evolutionary tracks in EXOFASTv2). This will double count the constraint and artificially lower the reported uncertainty.
        a. unless you know a certain parameter is well-constrained independently of the fitted data, you should not set the width

6. Changing starting values is wise. Adding penalties should be done sparingly and with thought and care.
v. The fourth entry is a lower bound.
1. Any model that steps outside of this bound will be rejected.
2. "-Inf" will be ignored.
3. Beware of biasing parameters away from these hard boundaries, like a Lucy-Sweeney bias against e=0. These should be physically justified, independent of the data sets being fit.
vi. The fifth entry is an upper bound.
1. "Inf" will be ignored.
d. If you have a previously completed fit, you can generate a new prior file with mkprior.pro that will start at the best fit among all links of all chains
i. mkprior, filename='idlsavefilename.idl', priorfilename='new.priors'
ii. This can dramatically speed up future fits
4. Prepare the fluxfile (optional, and skipped in the HAT-P-3b example)
a. The flux file contains the broadband magnitudes for the host star
b. This is used to constrain the bolometric flux of the star. When coupled with the distance (from Gaia DR2), it is a powerful constraint on the stellar radius
c. Can also constrain the temperature of the star
d. Most catalogs quote optimistic absolute precision -- be careful
e. This file can be generated with mksed.pro, from trusted catalogs and appropriate systematic error floors
i. mksed, 'starname', 'star.sed', ra=ten(0d0,0d0,0d0)*15d0, dec=ten(0d0,0d0,0d0)
1. 'starname' is the name of the host star. If it's recognized by simbad, it will be resolved (and ra and dec are not required).
2. 'star.sed', is the output fluxfile that will be the input to EXOFASTv2
3. ra and dec are the J2000 RA and Dec of the host star, in decimal degrees. The closest match to this RA and dec is used to grab the catalog magnitudes
f. This file can be generated by hand, but it is not recommended.
g. Do not use if the phometry is blended
h. Be sure to add a prior on parallax and an upper limit on Av in the prior file if you are using a flux file
5. Run the fit
a. Create the "fit_planet-name.pro" file. Use the HAT-P-3b example (or another example that is closer to what you want to do) as a template. Then, at the IDL prompt, type
i. Fit_planet-name
b. It's good idea to do a short (maxsteps=1000, nthin=1) test run to make sure everything completes correctly and there are no unconstrained parameters that cause the program to crash or make the results unusable.
c. EXOFASTv2 common arguments

i. NPLANETS
    1. The number of planets you want to fit in the system
        a. Must match the number of plan
ii. TRANPATH
    1. A string with wildcards that matches all the transit data files
iii. RVPATH
    1. A string with wildcards that match all the rv data files
iv. PRIORFILE
    1. A string that specifies the name of the prior file
v. FLUXFILE
    1. A string specifying the name of the flux file with broad band magnitudes of the host star
vi. PREFIX
    1. A string that is pre-pended to all output files.
    2. This is intended for you to keep track of the fit
    3. The PREFIX can include a relative or absolute path, which will be created if it doesn't exist. EXOFASTv2 has a lot of outputs and putting them into a subdirectory is a good practice to help you keep them straight.
vii. MAXSTEPS
    1. The maximum number of steps the MCMC fit will save
    2. For a publication quality fit, MAXSTEPS should probably not be smaller than ~1000.
    3. Large values may easily require a massive memory footprint (10+ GB) depending on the number of fitted and derived parameters. Take note of the warnings about memory usage and be aware that ignoring them may result in the program crashing at the end of a very long run.
viii. NTHIN
    1. (NTHIN-1)/NTHIN steps will be discarded to save on computer memory. Because adjacent steps are highly correlated, throwing away large numbers of steps (by setting large values of NTHIN) does not have a major impact on the quality of the fit. However, to the extent your computer can handle it, smaller values of NTHIN are better.
ix. CIRCULAR
    1. An NPLANETS boolean array specifying which planets should be assumed to be circular.
    2. If you're fitting 2 planets (NPLANETS=2) and you want the first to be circular and the second to be eccentric, set CIRCULAR=[1,0]
    3. If CIRCULAR is not specified, all planets are modeled with eccentricity.
x. FITRV

1. An NPLANETS boolean array specifying which planets should be modeled with RVs
2. If you’re fitting 2 planets (NPLANETS=2) and you want to fit RVs to both, set FITRV=[1,1]
3. If FITRV is not specified and RVPATH is supplied, RVs are fit for all planets. If not specified and RVs are supplied, no RVs are fit.
4. This will typically be either all 0s (no RVs) or all 1s (RVs), but there may be a mix if some planets are too small to have RV signals and you don’t want to include a bunch of poorly constrained parameters in the fit.
5. Note that setting it FITRV to 1 for a small transiting planet with an undetectable RV signal will still result in a robust upper limit on K/planet mass (but may add significantly to the runtime).

xi. FITTRAN
1. An NPLANETS boolean array specifying which planets should be modeled with transits
2. If you’re fitting 2 planets (NPLANETS=2) and you want to fit transits to the first, set FITTRAN=[1,0]
3. If FITTRAN is not specified and TRANPATH is supplied, a transit is fit for all planets. If not specified and no transits are supplied, no transits are fit to any planets.
4. When setting FITTRAN to 1 for a particular planet, make sure at least one transit of this planet has been provided.

xii. LONGCADENCE
1. A flag that, when set, will model the long cadence smearing of the Kepler/K2 mission.

xiii. DEBUG
1. A flag that, when set, will plot the data and model to the screen at each step
   a. This is very helpful when refining the starting values in the priorfile

xiv. VERBOSE
1. A flag that, when set, will print extra information to the terminal at each step, like the penalties for each

xv. YY
1. A flag that, when set, will use the YY isochrones to constrain the stellar properties
   a. http://adsabs.harvard.edu/abs/2001ApJS..136..417Y
2. MIST is generally preferred, but YY may be desired for some cases
3. Do not use YY isochrones for low mass stars (< 0.5 msun)
4. Only use one of YY, MIST, or TORRES.
   a. Set the NOMIST flag when using YY.

xvi. TORRES

1. A flag that, when set, will apply the Torres relations to determine the stellar mass and radius based on logg, Teff, and [Fe/H]
   a. http://adsabs.harvard.edu/abs/2010A%26ARv..18...67T
2. An empirical relation based on EBs
3. Do not use for low mass stars (< 0.5 msun)
4. Only use one of YY, MIST, or TORRES.
   a. Set the NOMIST flag when using TORRES

xvii. NOMIST

1. A flag that, when set, will disable the MIST stellar tracks that would constrain the star
2. These apply to stars 0.1 msun < Mstar < 100 msun
   a. http://adsabs.harvard.edu/abs/2016ApJ...823..102C
   b. http://adsabs.harvard.edu/abs/2016ApJS..222....8D

6. Stopping a fit early
   a. Sometimes, it's helpful to stop a fit and use the results as is, even though it hasn't met the convergence criteria. This is especially true while exploring the fits like we'll do here.
      i. In the IDL terminal, type
         1. Control + c
         2. !stopnow=1
         3. .con
      ii. WARNING: Type the above carefully. A typo can mean you lose the entire fit and have to start over
      iii. WARNING: Do this only if you are sure you want to stop the fit; once stopped, a fit cannot be restarted.
7. Outputs
   a. Files are generated by EXOFASTv2 in 4 distinct stages of the fit. **The most important outputs are underlined and in bold:**
      i. Once at the start of the fit; filenames are preceded by "PREFIX.start."
         1. These show the starting model guess (derived from the prior file and EXOFASTv2 defaults) overlaid on the data
            a. **<u>transit.ps</u> - A multi-page postscript plot of the transit model. The first page has each transit (file) stacked on the same plot, offset by a constant. Subsequent pages are the detrended, phase-folded transits for each planet. Note that the phase folded model may have some fuzz due to different limb darkening for each transit.**
               i. This plot is intended to be publication quality, and is for simple fits (e.g., many transits one per file, of a single planet), but is likely to be inadequate for publication for complex fits (e.g., Kepler/K2 light

curves, multiple planets). In this case, users may wish to generate their own plots using the text files generated described in section 7.a.i.1.f.

b. **rv.ps - A publication quality, multi-page postscript plot of the RV model. The last page is the unphased model with the average velocity, gamma, (systemic velocity or instrumental offset) and any fitted linear or quadratic terms subtracted from both the data and model. If more than one instrument is used, a legend is produced in the corner, labeled by "telescope" from the RV filename. The first pages are each planet, phased to its period, with all other planets, gamma, and any fitted linear and quadratic terms subtracted.**
c. **mist.eps - A publication quality plot of the star with its MIST isochrone overplotted. The black point is the best-fit value, the red point is the corresponding model value. Only generated if the /NOMIST keyword is not set.**
d. **yy.eps - A publication quality plot of the star with its YY isochrone overplotted. The black point is the best-fit value, the red point is the corresponding model value. Only generated if the /YY keyword is set.**
e. **sed.eps - A publication quality plot of the broadband photometry and best fit SED. Only generated if FLUXFILE is given.**
f. There are also text files generated to help the user re-create their own versions of these plots. We will not need these for the Sagan workshop, but see exofastv2.pro for more information on these.
g. NOTE: To extract a single page out of a multi-page PS file, use:
psselect -p# input.ps output.ps
where # is the page number to extract.

2. These are intended for quality control only, not for publication
3. Often times, it is necessary to tweak the values in the prior file until these show a decent match between the model and the data.
   a. For example, the fit will fail if the starting transit model does not at least partially overlap the transit in the data

ii. Once after AMOEBA is done; filenames are preceded by "PREFIX.amoeba."
   1. These are identical to the "PREFIX.start." files, except they show the best-fit model, as found by AMOEBA instead of the starting model.

2. These are intended for quality control only, not for publication
3. If the models are not good fits to the data, the MCMC is likely to fail and you should refine your starting values, priors, and constraints. At a minimum, the MCMC is likely to take significantly longer to converge if this is not a good fit.

iii. **Once after MCMC is done, filenames are preceded by "PREFIX.mcmc."**
1. These are identical to the "PREFIX.start" files, except they show the best-fit model among all links of all chains instead of the starting model.
2. These are intended for publication.

iv. And once at the very end of the fit; filenames are preceded by "PREFIX."
1. **median.tex - The LaTeX source code for a deluxe table of the median values and 68% confidence interval, rounded to two sig figs in the uncertainty.**
    a. This is the most important output and summarizes the entire fit
2. **chain.ps - A postscript plot of each parameter as a function of chain link, 8 to a page. Each color is a separate chain**
    a. This is the most important diagnostic plot and should be checked after every fit (see quality control in the next section)
3. **pdf.ps - A postscript plot of each posterior distribution function, 8 to a page. Each color is a separate chain. The thick black line is the average of all chains.**
    a. This is a useful diagnostic plot and should be checked for bimodalities or errant chains after every fit, especially if some parameters did not pass the convergence criteria
4. covar.ps - A postscript plot of the covariances between each fitted parameter in a corner plot.
    a. This (publication quality?) fit is useful to gain intuition about the covariances inherent in the fitted parameters.
5. median.csv - A machine readable table of the median values and 68% confidence interval, rounded to two sig figs in the uncertainty.
6. log - A file containing everything printed to screen during the fit for later review (or remote monitoring)
7. mcmc.idl - An IDL save file that contains the stellar structure with the full chains for each parameter, including derived parameters, the chi2 at each link, and the index of the burn-in period, called MCMCSS (MCMC stellar structure).
    a. WARNING: This file includes both links and chains that have been automatically discarded for a complete record

and to enable future diagnostics. Including those in any inferences may bias your results.

b. This is valuable if you want to publish the posteriors, or generate outputs or diagnostics we do not support (e.g., a covariance plot of two parameters). Proper use of this file is both complex and undocumented. Please ask for clarification if you think you need it.

8. Quality Control
    a. The quality of the fit and runtime is proportional to MAXSTEPS*NTHIN
        i. After the end of the fit, you may see warnings about the Gelman Rubin statistic and not being “well mixed”. You have two options.
            1. Run longer
                a. Restart at best fit from this run (see mkprior.pro)
                b. Increase MAXSTEPS and/or NTHIN
            2. Decide to use it anyway, effectively relaxing the convergence criteria.
                a. Only after careful inspection of chains and pdfs (see below)!
                b. You can also tell it to terminate automatically with different convergence criteria
                    i. e.g., in the call to EXOFASTv2, set “MAXGR=1.1, MINTZ=100”
                        1. The Gelman Rubin Statistic compares the mean of medians vs the median of means of the chains. It is essentially a measure of how similar the chains are and asymptotically approaches 1 as they become identical. The more similar they are, the more likely they each represent the true, underlying posterior. All parameters must each have a Gelman Rubin statistic less than MAXGR (default is 1.01).
                        2. Tz is the number of independent draws. Since adjacent links in the MCMC chains are highly correlated, Tz attempts to calculate how many truly independent samples you have taken from the underlying posterior by dividing the total number of links by the correlation length. The larger the number, the more accurate the PDF. All parameters must each have a number of independent draws greater than MINTZ (default 1000). Note that 1000 random

draws from a normal distribution would lead to a statistical error of ~3% in the determination of the median a ~2% error in the determination of the 68% confidence interval. Those errors scale as 1/sqrt(Tz). (i.e., a Tz 4x higher means half the statistical error).

3. See Ford, 2006 for more details http://adsabs.harvard.edu/abs/2006ApJ...642..505F

b. Chain file (prefix.chain.ps)
    i. This shows the value of each fitted and derived parameter (and chi^2), for each chain (color coded) as a function of chain link number
    ii. There is a black vertical line that denotes the burn-in.
        1. Everything to the left of it is discarded before calculating the posteriors
    iii. The chi^2 plot (the top left of the first page) is the most useful
        1. If the chi^2 is still trending downward, restart the fit at the best fit of this run and absolutely do not trust the results.
    iv. The important thing to look for here is that, after the burn in, there is no significant evolution to any of the chains and no chains are outliers in any parameters
c. PDF file
    i. This shows the “Probability Distribution Function” (a normalized histogram of the values in the chain), or posterior of each parameter.
    ii. Each color is a separate chain
    iii. The thicker, black histogram is the average of all chains
    iv. The important thing to look for is that it is relatively smooth and any bumps are not dominated by one chain
    v. Also note any bimodalities or highly non-gaussian PDFs. These may be misleadingly summarized by the median and 68% confidence interval and may require special effort to summarize.

9. Troubleshooting
    a. It’s always wise to run a short fit (MAXSTEPS~100) to start, just to make sure things look ok
    b. Use the /DEBUG flag to plot the data and the starting point of the fit. The closer the starting model is to the data, the better behaved it will be.
10. Tips and Tricks
    a. The memory required is proportional to MAXSTEPS. If you’re low on memory, you can increase the quality of the fit by increasing NTHIN.
    b. For a publication quality fit, MAXSTEPS should never be smaller than ~1000.

11. For the Sagan Summer workshop, EXOFASTv2 is already installed on the AWS you'll be using. If you wish to install it on your own machine, see the directions on github: https://github.com/jdeast/EXOFASTv2/blob/master/README.md