11/07/22:

Papers:

<https://arxiv.org/pdf/2002.06234.pdf><https://arxiv.org/pdf/2111.10169.pdf>

General plan:

***‘1) Repeat the analysis of Ke et al but adding optimal weights (Fig1 of this paper*** [***https://arxiv.org/pdf/2005.00244.pdf***](https://arxiv.org/pdf/2005.00244.pdf)***). If we don't want to go into the messy luminosity-redshift relation that detemines the radial kernel of neutrinos, we can simply parametrise it, for example as 1/(1+z)^q or 1/(1+z) step(z\_q-z) where q and z\_q would be free parameters (z is redshift). There should be some value of q and/or z\_q that maximise the cross-correlation (or auto+cross) and that might correspond to the distribution of "neutrino-loud" AGNs with redshift. This can be done with many galaxy catalogues, and we'd have to find the sweet spot between depth (in redshift) and sky coverage.***  
***2) Repeat the analysis but with different tracers, for example shear. We can't use weights here but the hope is that these tracers would "naturally" follow (in redshift) the distribution of neutrinosources, for example AGNs.***  
***3) Work out the luminosity-redshift relation and do (1) properly. But hopefully we will catch the AGN-nu correlation already with (1) or (2), so let's keep this as a plan B’***

Downloaded dataset from <https://icecube.wisc.edu/data-releases/2018/10/all-sky-point-source-icecube-data-years-2010-2012/>

Wrote code that reads in all event data and combines into single astropy table then creates HEALPix map. Currently the masking of pixels below –5 degrees declination does not work.

12/07/22:

Fixed masking issue – turns out pix2ang and ang2pix accept declination angles between 0 and pi for radians but –90 and +90 for degrees. Fixed by working exclusively in degrees (not true anymore).

Updated code to also load in effective area tables, and plot HEALPix maps of chosen energy bands.

*[Code looks up each neutrino event in the effective area table for that year and finds its effective area. Adds all these effective areas to ICFull. Takes 45 min to run, so saves this table to chosen path, which is then simply loaded in next time (if boolean InitProcessICData in config set to False).*

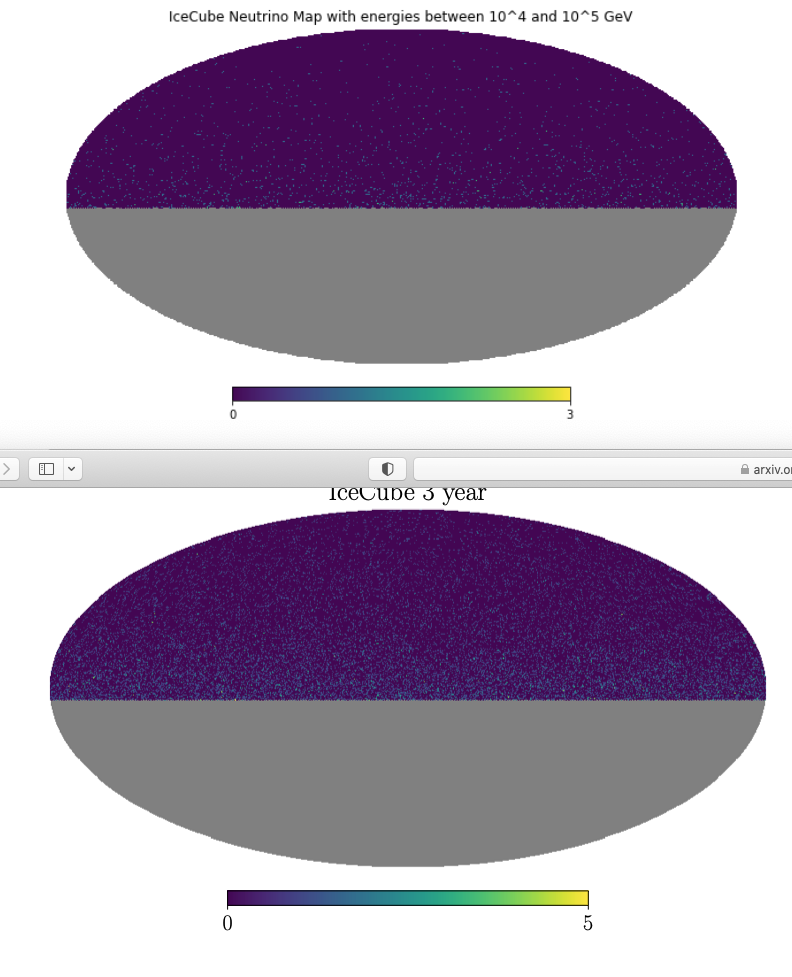
*Effective area table only has data down to 10^2 GeV, so must cut all neutrino events below this energy before searching for their effective area. Think this is okay as we are interested in high energy neutrinos.]*

Now no longer think need to find effective area for each individual pixel so *[this work]* is likely unneeded. New folder called ‘Scrap’ containing saved ICFull table with Aeff column, and screenshots of code in case required in future, but removed from code for now.

Added column in ICFull to show year the data was from, so corresponding Aeff table can be consulted if needed.

Started to write function to calculate the weighted effective area (equation B6 in Ke et al.) but not sure how to deal with the fact that different years have different effective area tables.

HEALPix map with same energy bounds also does not look exactly same as that in Ke et al. - not sure why as think using same dataset? (see below, top = my code, bottom = Ke et al.)



13/07/22:

Renamed ICFull to ICEventFull to distinguish better from effective area tables.

Programming function to take an energy and zenith angle (and year), and lookup the corresponding effective area in the table for that year. Effective area table only goes down to 2 GeV, however 268 events are below this value. Effective areas in this energy range are extremely small, so have assumed these cases to have an effective area of 0, but might need to check this. Similarly 15 events below 1.5 GeV, so are not in any of the energy bins of Ke et al. - is this okay?

Ke et al top right pg 3 says energy bins from 10^1.5 to 10^8.5 GeV, but bottom right pg 3 says three energy bins for likelihood function from 10^2 to 10^5 GeV. Unsure what the difference is.

Wrote weighting effective area function. Currently untested until actually useful for project. Still not sure what to do about different Aeff tables for different years, so for now just takes AeffTbl as an input.

Need to install NaMaster and xCell. Various problems:

* NaMaster brings up error with all installation methods (contacted Felipe on slack)
* xCell requires following packages that I can’t figure out how to install
* - pyccl (requires cmake which requries admin privileges to install)
* - camb requires GMP, MPFR, and MPC. Installed GMP, downloaded tar of MPFR so far.
* - have not tried any below camb. (see 14/07/22 for update, now fixed)

Wait until David is back next week for this.

Starting to write function to calculate equation 3 in Ke et al. (but with both a\_lms being neutrinos).

14/07/22:

Finished function to calculate c\_ls (equation 3). Learned this is a simple approximation, and NaMaster will do much better job when it is working. Also calculated c\_ls using healpy anafast function (also approximation), found similar coefficients.

Loaded in galaxy maps and masks – seem to be in different coordinates to those used so far as galactic plane is horizontal around equator. Will need to convert to coordinates we are using. - NSIDE for imported galaxy maps seems to be 1024 rather than 128.

Also need to change previous IceCube data from pixel counts to over/under densities. (Done)

Managed to install namaster. Still need to install xcell and rest of its dependencies, but will try namaster on its own for now. Computed cls using namaster package, and plot. Now have estimated cls from equation 3 in Ke et al., healpy.anafast, and namaster. All are similar in magnitude with slightly different features (mainly at low l).

Installed camb, now continuing down list of dependencies, will come back to pyccl last. Now downloaded all requirements for xcell, and the zip of xcell itself but not sure how to run.

Noticed galaxy maps have some pixels with underdensities below –1. How is this possible? From definition would require negative number count for that pixel.

Properly wrote code to load in galaxy maps and mask.

Realised calculation of over/under densities should only use unmasked pixels to calculate average as we are ignoring the others – fixed this and angular power spectra now look more similar to Ke et al.

15/07/22:

Decided it would be easier to convert neutrino maps to galactic coordinates than other way around. Done, but during conversion of coordinates some pixels in galactic coordinates have negative number counts of neutrino events leading to underdensities of less than –1. Possibly also the cause of underdensities less than 1 in imported galaxy maps? Converting in pixel space rather than harmonic space fixes this issue, but is also supposed to be less accurate? (<https://zonca.dev/2021/03/rotate-maps-healpy.html>)

Note that converting between coordinates like this is quite slow (especially as have to convert each pixel back to check declination for masking purposes), so cell block takes a while to run.

Implemented changing NSIDE of galaxy maps/mask to NSIDE of IceCube data as:

1. Believe this to be necessary for cross-correlation
2. NaMaster calculation for cls of galaxy maps taking very long time with their default NSIDE = 1024

Downgrading NSIDE for galaxy mask means that unmasked pixels (1) are averaged with masked pixels (0). Chose to treat any new pixels which are the result of at least one masked pixel as also masked. (Changed later to be consistent with rest of xCell).

Written code to calculate cls for auto-correlation of galaxies and cross-correlation of neutrinos and galaxies using namaster. Mask for cross-correlation is combined from both with:

* Unmasked + Unmasked = Unmasked
* Unmasked + Masked = Masked
* Masked + Masked = Masked

18/07/22:

Using kmeans package to find errors on cls using jackknife method. Also noticed cross correlation uses fields already initialised to compute c\_ls, however field initialisation depends on apodised mask, so need to generate fields new. Now fixed.

Need to compute covariance matrix and plot, then plot error bars on angular power spectra plot.

Plotted covariance matrix for icecube auto-correlation, need log scale to see anything, can see line for diagonal elements clearly except for highest energy bin.

Planning on changing IceCube mask code to have simply one mask for all maps to remove inconsistencies. (done)

Plotted covariance matrix for both auto and the cross correlation. Computing the cls for all the jackknife maps takes a long time so added boolean in config to skip this step and load from file on disk.

Plotted error bars on angular spectra but all errors for all correlations (with exception for IC auto-correlation highest energy bin) are extremely small. Not sure whether this is correct or not.

Calculated ‘chi-squared’ (is this the right name?) for each correlation and bin with sum of c\_l^2 / error on c\_l. All very large, on the order of 1000 - 10000, due to extremely small errors. Cross-correlation chi-squareds are on the order of 10-50 due to most cls being ~0.

19/07/22:

Fixed null hypothesis likelihood function (called ‘chi-squared’ above). Need square and factor of in denominator, was also previously missing the squared in numerator. Also change energy bins to three between 10^2 and 10^5. Therefore have to recalculate jk cls due to maps changing.

Turns out np.cov calculates covariance matrix, creating new function that calculates it this way to check my manual function. Gives same result but using np.cov now anyway just to be sure.

Need to add lmin functionality at some point as funky at low l (paper uses lmin = 50, should try different values and see how sensitive results are)

Meeting with supervisors: Should try generating results with 3 year, 10 year data (https://icecube.wisc.edu/data-releases/2021/01/all-sky-point-source-icecube-data-years-2008-2018/) and weighted aeff pixel counts (4 different sets in total). 10 year data also has uptime data – should check all this data is within uptime (when detector is working correctly). Aeff tables for last few sets of 10 year data are the same.

10 year data has slightly different formatting – different file names, column headers etc.

Focusing on adding weighted aeff to 3 year dataset for now.

Some aeff values are 0. Only 8 events where this is relevant (i.e. above minimum declination and within energy bins), all in lowest energy bin. Not sure how to proceed with this.

Plotted weighted effective area for each year and energy bin on healpy map.

For 10 year data, energy stated in events tables is muon energy, in aeff tables is neutrino energy. For 3 year data, energy stated in aeff tables is definitely neutrino energy, but unsure what energy is stated in event tables. Could be problem if muon energy like 10 year. Event data of IC79 2010 seems to be same for both datasets, with same energy values, so almost certain event table data is muon energy, and some sort of conversion needs to take place.

Changed null hypothesis to sum over energy bins but not redshift bins (yet).

Was missing sqrt when calculating errors from covariance matrix. Errors now more sensible, can be seen on plots, however large errors on cross correlation.

Added lmin = 50 for null hypothesis calculations. Makes little difference to result as low l contributions much smaller than high l. Inspecting sum elements for IC auto-correlation shows clearly that changing lmin will have little effect, also shown in below table:

|  |  |  |  |
| --- | --- | --- | --- |
| l\_min | IC auto | Gal auto (redshift 1) | Cross (redshift 1) |
| 0 | 10014.447209720296 | 4918.889036788837 | 42.54981493224812 |
| 25 | 9973.468593109259 | 4895.602551148814 | 39.131998686043445 |
| 50 | 9842.030647267207 | 4821.476710773432 | 36.83661664903365 |
| 75 | 9555.406688141473 | 4721.313955095384 | 35.35272450732407 |

Setting lmin = 50 as in paper seems good.

Todo tomorrow:

* Discuss what to do with aeff zero values with supervisors
* Check 10 year uptime (possibly in different notebook as only need once, not working on 10 year data yet).

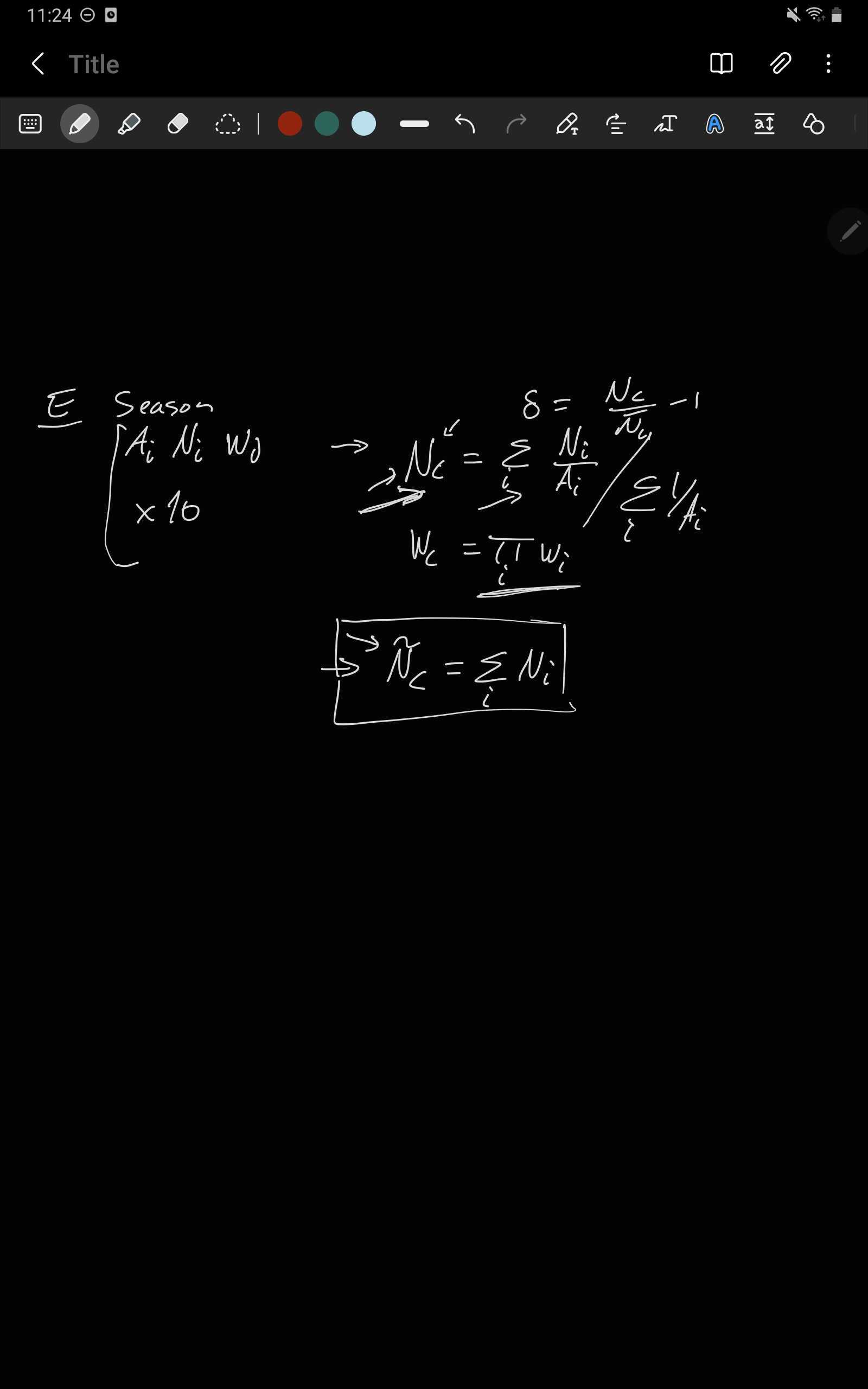
20/07/22:

10 year data files are listed in roman numerals. Sorting alphanumerically therefore does not put them in the right order. For now doesn't matter as sorts all files in same way.

Wrote code in separate notebook that loads in 10 year events and uptime data and checks to make sure all events are within uptime windows (and records those that are not).

Only one ‘bad time’ was found, in IC79\_exp.csv at 55506.37316921. The uptime window starts at 55506.37316922, about a millisecond later.

Supervisor meeting: For each energy bin, should generate count map, weighted effective area, mask (at x% of max effective area) for each season. Then combine count maps with weights for effective area and masks as described in:



Should also increase number of ls per band to 50 (now done). Also changed cross angular spectra plots to show positive in one colour and negative in another. This meant using scatter rather than line graphs as transitions between +ve and –ve points do not work for line graphs.

At the moment to generate weighted effective area maps, the declination of each pixel is run through the weighted area function. Instead, could first create table of weighted effective areas for each energy bin and season containing each cos(zenith) bin and then simply reference this table. This would hopefully be much quicker. Update: It is much quicker.

Creates mask for each energy bin, season weighted effective area map where pixels below a certain fraction of max weighted effective area are masked. Set to 10% for now but should test different values after lunch.

Varying fraction makes little difference to number of pixels masked (for sensible choices), so kept 10% for now. All masks are a straight declination cut here. Also now converted to galactic coordinates to match other maps and masks.

Noticed bug causing number of masked pixels in neutrino mask to always be 1, fixed and reran code.

Code now multiplies declination cut mask with all weighted effective area masks to create mask for IceCube neutrino mask.

Realised due to ordering of steps to create neutrino maps, far easier if weighted effective area maps remain in equatorial coordinates, only masks converted to galactic.

Tomorrow should upload a notebook with and without effective area weights used for comparison.

Figured out solution to files being out of order when sorted alphanumerically – must specify order of files to convert from alphanumeric to chronological in config section. Tested in uptime notebook for 10 year data and it works, but not yet implemented in main notebook.

Uploaded uptime notebook to git.

21/07/22:

Fast method of looking up Aeff by calculating the bin number first only works for 3 year data as 10 year Aeff tables have completely different format. Therefore changing it to simple brute force.

Generalised code for 10 year dataset up to cell which creates lookup table for weighted effective area map, although weighted effective area cells have not been tested with 10 year data.

Uploaded two versions of MainCode to git – with and without using Aeff weights.

Code to generate weighted effective area maps is now very slow due to having to find zen/dec angle in reference table with brute force loops rather than calculating bin number. To fix, instead of the reference tables having a row for each bin, will make them have a row for each different declination in hp map. Reference tables do take a while to generate so now have them save/load similar to JKcls.

Generating weighted effective area maps is still slow so moved save/load to this point, maps are simply loaded most of the time.

10 year data now works with code.

From supervisor meeting, in many places for loops are used where numpy array magic could be used, which would make the code faster. Going through and trying to implement this as much as possible. Done up to and including cell which saves/loads weighted aeff map data.

22/07/22:

Continuing to clean up code. Now done up to end of IceCube data section. Now up to start of angular power spectra.

Need to plot cross angular power spectra with errors in easy to read format for zoom. (done, saved as CrossPowerSpectra.pdf).

As 10 year data Aeff tables use simply declination for angle, and will likely be focusing on this data, changed use of zenith angle by default in effective area functions to declination by default.

Turns out healpy.anafast does not take f\_sky into account by itself – must divide by it afterwards. Now done and anafast method is much more similar to other two methods now.

25/07/22:

Comparing David’s code to my own:

Differences:

* D code interpolates weighted effective area between declinations to have smooth rather than jumpy weighted aeff maps. This also causes small region of sky around north pole to be masked.
* D code has additional energy bin from 10^5 to 10^6 GeV.
* When downgrading nside of galaxy mask, D code sets those pixels above 0.5 to unmasked, I set all below 1 to masked.
* D code has sixth redshift bin (map 0).

Made all these changes and reran.

Still not exact same results. I think the difference is in the namaster calculation of cls, although do not understand D implementation. D code does not apodise masks, and uses ‘workspace’ to calculate cls. Apparently this workspace accelerates the calculation as long as the mask does not change. Also noticed scale of over/under densities on plotting neutrino maps is different between codes so must also be difference before this point.

Plotted both David’s and mine results on same plot to compare in tomorrow’s meeting.

Now calculating ‘theoretical errors’ to check reliability of Jackknife errors. Only doing this for first energy and redshift bin for now. Calculation using namaster requires workspace of cross power spectrum to be given, so rewriting (all) previous namaster stuff to use workspaces. (also removing apodisation). (Done).

nmt.gaussian\_covariance gives error when inputting cls calculated previously – seems like need 384 cls (I.e. using 1 l per bandpower). Have asked David about this. Apparently can just use healpy anafast for this.

Done and plotted errors calculated by both methods next to each other. Very similar. JK errors slightly higher at low l, but theoretical errors slightly higher at medium-high l (in general).

Back to trying to find difference between codes. One thing is that D code never uses .rotate\_map\_pixel, which isn’t completely accurate – should try to eliminate uses of this tomorrow (I.e. rotate angles when building maps instead).

26/07/22:

Removed use of .rotate\_map\_pixel (in interpolation method at least – non-interpolation method is as a result broken). Fixed most of discrepancy in plotted neutrino maps. Rest was fixed by changing condition for lower bound of energy for each bin from > to >=. Ran several tests and am convinced both codes now give exact same results up to this point, but end results still different.

Computing cross correlation in my code was accidentally using workspace from galaxy auto correlation. Fixing this means the code now gives the exact same cl datapoints, although a couple of the errors are slightly different.

Figured out the difference is just due to different JK regions being drawn each time, so slightly different errors are being estimated.

Uploaded weighted effective area maps and event counts for each energy bin, season to github so synthetic data can be generated using them.

Need to write xcell mapper class for neutrinos. Following mapper\_NVSS.py closely, written by Felipe. Just need to make functions that create signal map and mask for now.

27/07/22:

Think David’s code for the weighted effective area function can only handle when fine energy bin bounds line up with wide energy bin bounds. I should code my version in the xcell mapper to be completely general.

Have created get\_mask and get\_signal\_map functions, and uploaded the class to a new branch in xcell github. Now need to create test\_mapper\_IceCube.py in order to test it.

Need to use flake8 to test formatting of mapper file. (Done – should repeat after changes are made).

Need to use pytest to run test mapper. Tests of \_get\_events and get\_mask work. Test of get\_signal\_map show that maps outputted as all 0. Don’t think this should be the case as although 1 event per pixel per season, using real aeff tables so should still be fluctuations.

Not sure how to fix, so will try to test aeff tables of all 1s (which if used with above WOULD create signal maps of all 0s), and hope this reveals an error. Need to generate such tables in order to do so.

Have written code to generate such tables, but not yet tested. Do rest tomorrow.

28/07/22:

Fake Aeff test works – output is aeff mask with only dec values cut, and aeff map with all non-masked values = 1.

Get\_signal\_map still returning map of full zeroes – think this actually should be the case due to dividing by normalisation but commenting out normalisation still returns zeroes, which is very weird (and wrong). Turns out error was just me. (wrong index – j instead of I). Also needed to create 1 event per pixel per season per energy bin.

All works now – except that get\_signal\_map using real events and effective area does not give same map as Jupyter notebooks. Max of first energy bin for example is in the 600s.

Also changed get\_signal\_map to just generate a specified energy bin rather than returning all.

Should make test script delete fake data generated as takes up a lot of space. (done). Real data still exists though, takes up fair amount of space, don’t know how to get round this.

Fixed get\_signal\_map (finally). Problem was in get\_events – due to how self.cat\_data was defined at the start, code thought all seasons after first had already been run and simply returned the data from the first season. Now fixed, and agrees with results from notebook. Need actual data to perform this test though, which takes up a lot of space so not sure whether should keep it in test script.

Ended up removing test for this reason, although kept in Nuxcorr git as still nice test to have as sanity check. Need to come up with new test now as previously mapper script was passing all other tests while still being broken.

Happy with tests, uploaded latest versions to git, and now moving on to papers linked in slack.

29/07/22:

Supervisor meeting: should look at ccl (core cosmological library), try to reproduce Angular cross-correlations.ipynb

Had trouble importing pyccl – first because couldn’t import camb. Changing boltzmann.py in pyccl code to not attempt this fixed this issue (as well as adding parameter to line setting cosmology to use). Then unable to import some functions from scipy.special. Commented out these imports:

halos/profiles.py - stop importing from scipy.special

nl\_pt/Tracers.py - stop importing from scipy.special

Now however unable to import tracer, so will instead try reinstalling scipy.

Reinstalling with conda didn’t work, reinstalling with pip did. Removed comments in two scipy imports above, now works. Version of pyccl installed does not include two of the tracers used in the example code. Don’t think will need these in project, but are used in example code so trying to update pyccl anyway. Used ‘pip update pyccl’ to only update pyccl but seemed to update a lot of files/packages anyway.

Now can no longer launch jupyter notebook through anaconda (but can through terminal), and am unable to import numpy. Trying now to reinstall jupyter notebook on anaconda. Can’t seem to uninstall it. Trying to downgrade to previous version. Also doesn’t seem to work. May need to try reinstalling anaconda navigator on Monday, although am unsure whether installed packages will be lost.

01/08/22:

Reinstalled anaconda navigator, which went fine. Do need to reinstall all packages, but most are fine. Namaster, pyccl, and camb taking forever on solving environment step of conda install, other install methods not working either. Letting conda install of pyccl simply run now.

Ran command for 2 ½ hours, still hadn’t finished. Cancelled and instead created new environment ‘InternshipEnv’, and installed all useful packages there. This now works.

Followed through the recommended ‘flow’ of example pyccl notebooks, and recreated Angular cross-correlations.ipynb as suggested. David in all day tomorrow, will meet to go over theory prediction.

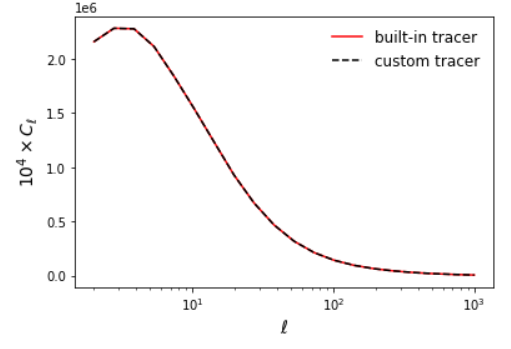
02/08/22:

Familiarised self further with Angular cross-correlations.ipynb and GeneralizedTracers.ipynb. One thing not sure on is how angular power spectra are calculated – only redshift/radial distances are inputted, not angles in sky? Think it does this through the defined power spectrum. (By default this is the non-linear matter power spectrum, I.e. underlying matter overdensity). In angular cross-correlations for example, this is found from the inputted halo mode for each tracer.

Supervisor meeting: Need to update E bins in xcell. Play around with ccl tracer – create CIBTracer for neutrinos. Use galaxy redshift distributions to to get galaxy tracers, get cross-correlations.

Having trouble running mapper\_IceCube now – it can’t find namaster now it’s in different environment. Fixed by just switching terminal to correct environment. Turns out environment was missing a couple packages used by xcell, now added and it works. Passed tests with new E bins, and pushed to git.

Wrote code to create custom IceCube tracer using parameterised equation A(1+z)^q where A and q are free parameters. For now put both in config at top as single values (and for now both equal 1), but will likely need something cleverer in future. In theory with A = 1 and q = -1, should be exact same as CIBTracer. Testing this now with radial kernel redshift range of [0.05,6].

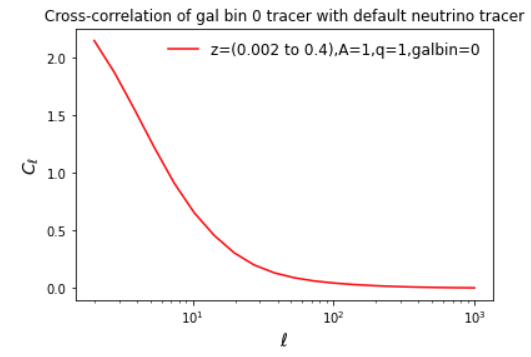


This works.

Put generating neutrino tracer inside function so it can be called with different A, q values in the future.

Loaded in galaxy redshift distributions for each redshift bin, created a tracer for each with bias values constant with redshift taken from arxiv:1805.11525. Changed default range of z values for neutrino tracer from [0,0.4] to [0.002,0.4] as this is the redshift range of the input distributions.

Computed cross-correlation for first redshift bin and default neutrino tracer (assuming underlying matter overdensity for power spectrum) Does not have ‘hook’ feature at low l like above, however increasing z\_min to 0.05 to match above does recover this feature.



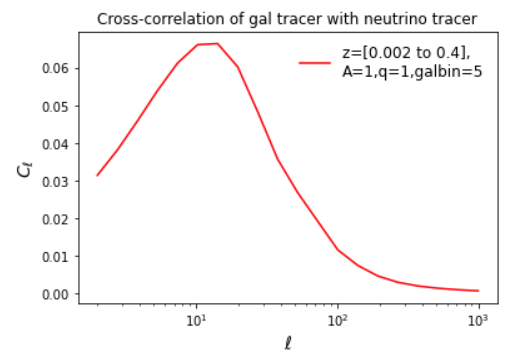
Unsure what direction after lunch:

* Test different z ranges
* Test different values of A, q
* Use halo model for power spectra (as in Angular cross-correlations.ipynb)
* Test different galaxy bins

Decided to first stick with one galaxy bin and test different A,q values. Range of A=(0,5] step 0.5, q=[-5,5] step 1, I.e. 10 x 11 = 110 total.

Higher A means higher cls, although shape of graph appears unchanged for constant q. This is as one would expect for a constant of proportionality. Hard to tell change in q due to large number of plots – changing A values to [1,5] step 1 and q values to [-5,5] step 2. Seems like slightly higher cls as q increases, and initial drop becomes slightly more curved than straight, but these changes are very small, hard to tell by eye without quantifying somehow.

Next test different galaxy redshift bins at A = q = 1 (at least initially). For increasing bin number, peak in cls appears and moves to higher ls (smaller angular scales). E.g. for bin 5:



Repeated broad ranges of A,q for galaxy bin 5, found very similar patterns, except the increase in cls with higher q values was much more exaggerated.

Should now test impact of varying redshift range of neutrino radial kernel. Will do this at A=q=1 as before, but with galaxy bin 5 as this has the most features to be able to see differences. Will test lower bounds of 0,0.001,0.002,0.005,0.01,0.05, and upper bounds of 0.35,0.4,0.6,1,2,6.

Changing bounds seems to have very little effect. Cls increase slightly with the upper bound but stay constant after 0.6. Consistent with any upper bound above 0.4 not mattering due to galaxy data not going above this. Actually found out galaxy data goes up to 0.6 in bins 1 through 5. Changing previous plots to this upper bound has no major effect.

Repeating this for galaxy bin 0, which may show differences due to lower bound more clearly, find now no change due to varying upper bound, but hook feature appears more prominently at higher lower bound. Likely due to low redshifts being cut out, effectively increasing average redshift of this bin (a bin 0.5 perhaps) and so this hook is just the peak seen in higher bins starting to appear here as well.

In conclusion will stay with redshift range of [0.002, 0.6] as this seems to include all data with no adverse effects.

Apparently halo model not required. David will send over xcell power spectra shortly.

David sent over power spectra in ‘sacc’ format as well as a notebook showing how to read data. Next task will be to compare xcell power spectra with those calculated in MainCode.ipynb to ensure they match.

03/08/22:

Plotted power spectra from xcell in same format as MainCode.ipynb. A few individual datapoints are quite different but overwhelming majority agree. Main difference I can spot is the first cl in each plot. For MainCode.ipynb, error bars are mostly quite small, and all but one is positive. From xcell error bars all very large and about 50/50 are positive/negative (perhaps as a consequence of these larger errors).

Read papers sent by Felipe on number counts in galaxy surveys (mostly just skipped to conclusions).

Made plots comparing error bars calculated using Jackknife, Namaster, and xCell. XCell errors appear only slightly larger for the first datapoint than Jackknife on a log scale, but in reality are on average about twice as large. Also made plot comparing power spectra calculated by MainCode and xCell.

David will investigate why first l bin shows these differences. In the meantime can continue using xcell power spectra and attempt to constrain (A,q) parameters. To do this, need likelihoods apparently, need to talk to Felipe about this.

David will try to find resources to help do this/explain how to do this. In meantime read through other (tangentially) related papers and cleaned up code/file system a bit.

04/08/22:

David has sent notebook about parameter inference. Implementing techniques to constrain (A, alpha) for each cross-correlation in xcell. For priors, am choosing top hats with mod(A) <= 0.1, as examining theoretical and xCell power spectra, A must be significantly less than 1, and mod(alpha) <= 10 to avoid silly answers. Will see how long this takes to run/ quality of output and possibly adjust accordingly.

Running MCMC takes a while, but only need to do it once – afterwards can load from disk. Written code for various plots in meantime. Corner.corner for corner plots seems to only display errors with quantiles of [0.16,0.5,0.84], despite docs saying this can be changed, so using these quantiles for now.

Turns out pickle doesn’t like the sampler objects, so instead first grabbing the chains (discarding the first 20% as ‘burn in’) and saving these instead.

Have noticed while testing various plots that the example mcmc notebook has fairly large range in the posterior distributions it spits out. The distribution can become negligible anywhere from A =4 to A=10.

MCMC finished running. A can probably put upper bound on (or at least upper bound on mod(a)). Alpha probably not – likely need larger priors. Will now share findings on slack. For whatever reason this particular plot is just a blank image when saved as pdf, so saving as png instead.(Think not blank, just takes very long time to load).

Ran MCMC for just energy bin 1, redshift bin 3 (chosen as it seems fairly typical in above plot) with the priors on alpha changed to mod(alpha) <= 100. Seems upper bounds on mod(A) and alpha will be possible. Prior of 100 is larger than required (in this case), so trying 20 for this bin, then will try for a couple other bins to make sure it is appropriate before committing to fully running.

After further tests decided to just use mod(alpha) <= 100. Running again for all cross-correlations.

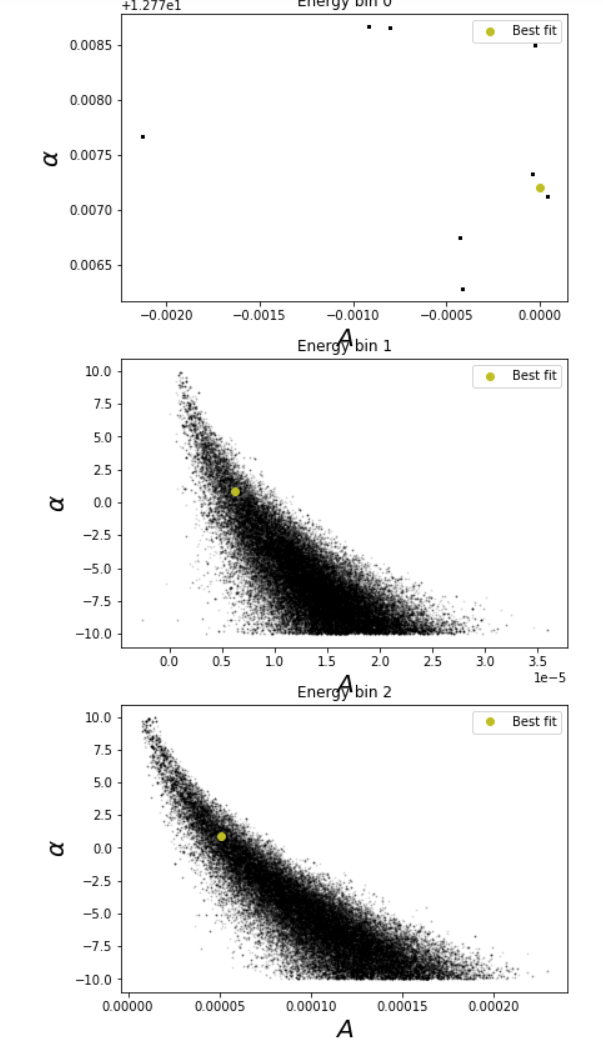
Apparently next step is to combine all redshift bins for each energy bin. Not sure how to do this, have asked for further clarification. Turns out as simple as changing data vector from len 7 vector to 6x7 matrix. MCMC for mod(alpha) <= 100 nearly done, so will wait for that and check sensible before implementing this.

With this prior high redshift bins have very unconstrained As. Will combine all redshifts first, then adjust priors if needed.

Not sure how to handle the inverse of the covariant matrix as it is different for each redshift bin. Based on adding errors in quadrature, maybe just add covariance matrices? Assuming this for now, but will make sure to ask David later. This does not work, as resulting (d-m)^T c^-1 (d-m) is then not scalar as (d-m) is now 7x6 rather than 7 (breaks scipy minimize function). Maybe another sum of the resulting matrix, but really not sure. Again assuming this for now.

Running MCMC for this. First bin had a warning and finished within a few seconds – very suspicious. Second bin taking very long time. Rest of notebooks (basically just plots) updated to handle output of MCMC, so essentially just waiting on this. Dubious it will work now (16:51 as of writing), but will talk with David either at end of today or tomorrow morning to check/correct assumptions. Second bin finished at 16:53 after ~30 mins, will wait today until third bin finishes, just to see output (although don’t think it will be correct given first bin).

Another possibility for the chi^2 could be just summing individual chi^2 for each redshift bin? Seems slightly more likely in hindsight, but too late to change for this run anyway.



First bin failed as expected, but next two worked well. Will check why first bin failed and chi^2 assumptions tomorrow.

05/08/22:

Supervisor meeting (long): David wrote code showing how to handle chi2 for combined redshift bin case. Will send it over, can then incorporate and run MCMC. Will use no A prior and mod(alpha) <= 10 (for now at least). Also discussed synthetic data, will be sent over soon, can then run xCell pipeline on it. David showed how to edit config yaml file and sent the necessary terminal commands to run xCell on slack.

Not yet received said code for chi2. Tried to reproduce it by memory, only issue is runtime warnings in best fit, but still obtain best fit values. Will try MCMC now, and hopefully compare my code to David’s later. Few minutes later received code from David, reconciled differences (largely superficial), and ran MCMC. Noticed while running that best fit for energy bin 0 is –9.9999999995 (give or take a few 9s), so likely running into bottom of prior. Will see what output is like but may need to expand prior (although maybe not – if we only expect to get upper bound on alpha anyway, would we not expect best fit to always be near bottom of prior?).

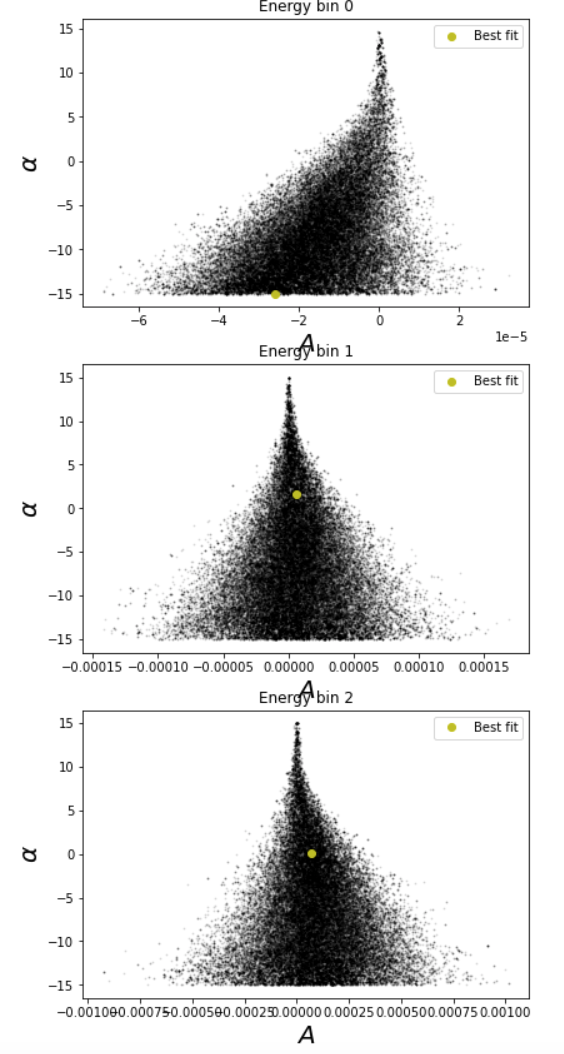
Finished running, plots all look good, different shapes to those above, which makes sense due to corrected chi2 now. A values have upper and lower bounds, but are consistent with zero. Alpha values possibly have upper bound but posterior distribution does not completely tail off at alpha = 10 for any energy bin. Therefore testing now with alpha\_prior = 15.

There is a warning at start of MCMC about ‘invalid value encountered in double scalars’. Believe this to be related to chi2 being very large so e^(-chi2/2) being approximated to zero. Only happens at start, so believe this to be due to start position of walkers. They start at a random position inside a square in parameter space of 0.001 side length. This means only a small difference for alpha, but means A is nowhere near the best fit value. Restarting current run to try to adjust this accordingly, although only small issue and don’t expect it to make a huge difference. Changed (0.001, 0.001) to (1e-9, 1e-4). Warning still appears (although weirdly only when restarting and running all, not when stopping mcmc and rerunning cell), however believe these to be better choices anyway, so may as well keep them.

Looked in IceCube.yaml file while waiting – galaxy bins past first seem to have extra files I’m not sure I have. There is ‘data\_catalog’, ‘spec\_sample, ‘mask’, and ‘star\_map’. Think I only have ‘data\_catalog’ and ‘spec\_sample’, so will need to ask about this.

Another thing from supervisor meeting is that we will have a proper talk next week about best priors to use as choice is non-trivial, so no point trying to ‘optimise’ it now – will likely leave it after this run unless output is especially crazy.

MCMC finished, pushed code and plots to git and shared results on slack. Output looks sensible (yay!).



After further consideration, warning may be arising when a walker starts outside of the prior range? Would explain why it only happens to the first energy bin, as this is the only best fit pressed up against the edge of the allowed parameter space, so the random addition to find the starting positions has ~50% chance to take each walker into ‘forbidden’ position. This would cause chi2 to be infinite, so likelihood to be exactly zero. On the other hand, surely walker meanders into forbidden space during MCMC as well before learning not to? Again, don't think it really matters, so no use investigating further or trying to fix.

Believe next step will be running synthetic data through xCell pipeline once it is ready.

08/08/22:

Currently trying to figure out how to download additional galaxy map files needed for xCell pipeline. Downloading two linked files on website <http://ssa.roe.ac.uk/WISExSCOS> just downloads single csv files of 5 and 15GB (not very helpful). Reading through documentation on SQL database now but likely far easier to ask David to send them over. Also now think want this page <http://ssa.roe.ac.uk/TWOMPZ.html>, which doesn’t even have any links, just SQL database. Definitely easier to just wait for them to be sent on slack (if even required, maybe xCell won’t care as not needed for cross-correlations).

David mentions adding ‘regularization’ to alpha priors, will go through in meeting later. Read through this <https://bjlkeng.github.io/posts/probabilistic-interpretation-of-regularization/>, explaining what L1 and L2 regularization are. L1 favours ‘sparsity’ where as many parameters as possible are (close to) zero. L2 favours situations where all parameters are small.

Notebook for synthetic data sent over. Generated synthetic data, split into tables for each energy bin, redshift, and year, as well as one combined table for each energy bin. Three lots of these, for the cases of f\_astro = [0,0,0], [1,1,1], and [0.0022, 0.012, 0.15]. Technically this means all data is saved twice, once in individual tables, and again in each combined energy bin table. Unsure whether individual tables will be required for now, so will leave as is.

Noticed for any given table, the number of events decreases significantly as f\_astro increases. Am unsure whether this is intended behaviour. Seems events are generated in all regions of sky, not just unmasked regions, which could be reason for this? Maybe unmasked Ntot doesn’t change? Felipe will investigate this while...

David added section to ParameterInferenceExample.ipynb about regularisation of alpha. - Need to add ‘volume prior’. Done, and now running with alpha priors of 10.

Trying to run real data through xCell pipeline just to test it works. Despite switching to new InternshipEnv conda environment, script is unable to import numpy, ‘ModuleNotFoundError’. Running a ‘testscript.py’ through terminal is however able to import numpy (in both environments), so not sure what issue is. Contacted David, who is not sure either.

MCMC finished, now rerunning with alpha priors of 15. Finished. All alphas unconstrained in both cases, A values very similar for both prior choices. Not enough time left today to try another choice.

09/08/22:

Weirder still, can import and use numpy using python3 in terminal, import numpy using other ‘test scripts’, and yet just this specific xCell script is failing. This is the case in both environments, and the xCell script runs into an earlier error unable to import namaster in the base environment, so do not believe it to be an issue with the wrong environment. Running problem script manually in terminal is also able to import numpy (although crashes later due to not being ran correctly). This is also the second time running through the script that it is unable to import numpy, so it must work fine the first time.

Have been running alpha prior of 50 in meantime. This seems very large, and has been spitting out many errors (although no crashes, believe still working fine), so will not go larger than this. As expected, plots are bad. Trying now to decrease alpha prior to 5.

Supervisor meeting: xCell is fixed! Need to compute constraints on A for each energy bin given fixed alpha of –2,0,+2 (so 9 in total) in order to test dependence of constraints on A on alpha. Then need to calculate pte of chi2 of each power spectra (for each energy bin, redshift bin, and combined to get for each total energy bin) using chi2 from scipy.stats.

Have stopped alpha prior of 5 code – don't believe it will be useful anyway. Running fixed alpha of +2 initially, then will change to others in turn. In theory code can now handle alpha either as a fixed or free parameter although the latter case has not been tested.

Calculated SNR of each power spectra (or rather how many sigma each chi2 is from the mean, assuming normal distribution – assume this is equivalent?). For combined energy bins, first two have ‘negative SNRs’, meaning their chi2 is below the number of degrees of freedom, of ~-1.4, third energy bin has SNR of ~0.1. None of these or the individual bin SNRs are significant. Code finished running for fixed alpha = 2, now repeating for 0.

xCell now not liking galaxy data – need actual galaxy catalogs, not galaxy maps, David sent these over, working now. First command finished successfully. Second to compute covariances is giving an error on every matrix saying it is singular. Don’t think anything is being outputted either, so will likely have to check with supervisor, but about to go to lunch so will leave running anyway. Should test whether cls are sensible by loading in and plotting in python – this could be likely reason of error.

All fixed alphas (-2,0,+2) now completed. For each energy bin the constraints on A are roughly the same, increasing slightly (generally just under a factor of 2) from alpha of +2 to –2.

xCell cov command now finished, as expected nothing outputted, all matrices gave error as they were singular. Examining calculated cls shows sensible results for ebin 0, zbin 0 but for all others only the first data point has been calculated. For this first bin the cls match prior calculations exactly, and although the polarity of the other first data points are correct, they seem to all be of order 10^-323.

10/08/22:

Running cls command again. Noticing that for first bin only, ‘setting the output map dtype to [dtype('float64')]; setting the output map dtype to [dtype('float32')]’ is printed. Could be cause of subsequent errors in following bins? Testing first few cls that have been outputted shows exact same pattern is happening. Now finished running, get exact same ‘results’.

Found that clearing reruns directory between bins seems to work, but obviously not practical. Tried moving output directory so that it did not contain reruns directory as in default yaml file, did not work. Changing rerun path in yaml file to ‘’ also did not work.

Going to work on slides as suggested by David in meantime. ***‘ok, since the synthetic data may take a while, I'm thinking, why don't you start preparing a few slides containing both the results you currently have (power spectra, corresponding constraints on A and alpha), as well as the theory background (i.e. what we would expect the radial kernel to be, how we model it, and also this regularization of alpha)’***

Finished slides, pretty happy but will look over tomorrow with fresh eyes.

Saving and loading ‘d’ in \_rerun\_read\_cycle causes it to break even the first bin, even though the arrays seems to be exactly the same before and after.

11/08/22:

Changing output dir to inside xCell folder made no difference, moving back now.

Forcing xCell to regenerate IceCube mask only, and keeping saved 2MPZ mask works, but not the other way around. Clearly then the IceCube mask is the problem child. Returning IC mask as float32 instead of float64 works (huzzah!).

Running xCell commands on real data now, and David has asked to do parameter inference on 2 sacc files sent over, both including and not including first datapoint, and as combined and split energy bins (but only 1 energy bin, fixed alpha of –1, so 28 plots in total). Doing this now.

XCell to\_sacc command failed – no module sklearn. Installed scikit-learn, now works. Will test output in a sec. In meantime, have coded parameter inferences requested by David. Four notebooks are running MCMC currently, one for each sacc file (2 in total), and for combined and split energy bins. After each has finished will need to change boolean in config to repeat without first cl data point. Seems to be very slow (assuming something to do with the fact that four are running).

Tested xCell output – same as that sent by David!

No\_split notebooks finished, had to change a couple things in plots, will have to do same for split notebooks when they complete. Rerunning no split now without first datapoint. Update: Now running both splits without first data point – these are the last two to run.

Synthetic data catalogs have been sent by Felipe, need to run both through xCell pipeline. Files are in fits format, so need to do some processing before putting into pipeline. Trying to run synthetic data notebook and am encountering error ‘hitmap’ is not defined. Unsure what ‘hitmap’ is supposed to be, so have messaged Felipe on slack, who sent hitmap function over.

Catalogs are tables, going to correct column headings, make sure energy eg is in format understandable by pipeline. Saving just for redshift bin 0 as originally sent by Felipe.

Saved catalog in required format in respective directories in xCell\_output. Running synth\_bestfit through xCell pipeline now.

Ran both sets of synthetic data through xCell pipeline, then both sacc files through xCellPowerSpectra.ipynb and saved resulting plot in respective xCell\_output directories. Synth\_bestfit is mostly noise as expected, fullastro shows strong correlation for first redshift bin (and 2nd/3rd due to overlap in redshifts) as expected.

Tomorrow will need to upload/share results from today (I.e. MCMC results and synthetic data power spectra).

12/08/22:

Outputted constraints on A in each MCMC case to text file. Pushed plots to git and sent this text file on Slack. Also shared synthetic power spectra at same time.

Tried running synth power spectra through SNRCalculations.ipynb. For many bins, including all three combined energy bins in full astro case, pte is approximated to 0, and SNR/sigma thus unable to be plotted.

Will talk more about synthetic power spectra on Monday when Felipe is back in.

Created plots showing global and per-bin constraints on neutrino density rate and luminosity density as a fraction of the critical density for both cases including and excluding l=2. These are key plots.

So far plotted x axis using midpoint between two edges of redshift bins. Instead want to use mean of redshift distributions – coding this in now. (Done).

Now adding last couple days to slides.

From when David sent this data: ‘***The files contain only the 6 cross-correlations with a single energy bin (you don't need to know which one, I'll explain later).’*** Maybe ask about this on Monday which energy bin it is, whether it needs to be repeated for other energy bins.

15/08/22:

Showed slides to European collaborators in morning. Now need to add more redshift bins from other surveys. Other surveys obviously have different masks, so adapting David’s code ‘GetFluxCls’ to cope with multiple galaxy masks.

Adapted code and generated fits files for number and energy flux. Running all parameter inference notebooks for including l=2 at the moment. Will work on slides in meantime.

Including l=2 finished running. Z bins 10,11 didn’t work at all (QSO bins). Bin 9 is also dubious. Believe likely to be due to l=2 datapoint, so will run notebooks without l=2 anyway and see what happens. (Best fits are not a good sign – inf for bins 10, 11 as before). Unlikely to finish today unfortunately.

Pretty much finished slides – will need to update ‘money plot’, maybe add a slide or two for new results. Other things could be slide at start giving outline of presentation, and slide at end for summary, but these would probably be best done on Wed afternoon/ Thu morning once rest is finalised.

~40,000 events in energy bin used for flux stuff/’money plot’ (possible q at presentation).

Leaving last few bins to run overnight, will see results tomorrow morning.

16/08/22:

Same issue with nol2 notebooks. Will combine into ‘money plot’ anyway to show to supervisors.

‘Money plot’ looks awful – bins 10,11 far too high in number flux case, not as bad in energy flux case (although still clearly too high), but then bin 9 is far too negative.

Apparently don’t need MCMC for one linear parameter like this case, so instead changing it to work analytically. Also changing it to just one notebook (instead of effectively eight), so should run much easier and far quicker. David sent updated example parameter inference notebook, showing how to do this analytically.

Still having difficulty with last two bins - ‘t’ (basically the model with A=1) for each of these has 0 for every data point. Issue is ccl.angular\_cl of neutrino tracer and galaxy tracer returning all zeros, despite neutrino tracer being the same for all bins, and the (auto-correlation of the) galaxy tracer looking very similar for all bins. Will continue for now with just first 10 bins, come back to this later.

Made constraints plot for first 10 bins, DELS\_3 has large uncertainties and is very negative in some plots, but may well be that this is the ‘correct’ result for this bin. Not sure how to fix QSO bins, have contacted supervisors on slack.

Issue was neutrino radial kernel only being calculated up to z=0.6. Changed this to z=4 (the highest redshift in the highest redshift bin), and reran. QSO bins now work, and DELS\_3 looks far more normal. Still huge uncertainties on QSO bins though, and when l=2 is not included, they are very negative. Increasing max z further has no effect.

Adding new results to slides. Adding extra slides: extra ‘money plots’ and atmospheric neutrinos.

Considering adding effective area slide in – otherwise will just explain it and point out related mask on maps slide.

Probably also need slide explaining cross-correlation/angular power spectra (or at the very least work out how to explain it).

17/08/22:

Added to maps slide explanation of cross-correlation with some fancy equations. Will likely just explain effective area on this slide as well, point out blob in northern hemisphere due to aeff mask.

Rewrote parameter inference notebook to only need to be ran once to generate full A\_constraints.txt. Much easier and more intuitive to use now.

Likely have meeting with supervisors in afternoon to go over presentation, so will practice (in head) now and time it. Took just under 20 minutes, but will be much quicker when practiced a bit. Will be fine for time. Not a fan of summary slide, will likely change/remove.

Currently neutrino constraints plot with just first 6 bins has blue bar calculated for all 12 – should fix this at some point. (Done.)

Now checking how results change at different dec cuts. Testing –5 as before now to make sure changing dec cut works. Looks same as before (good sign). Repeating now with 0. Looks similar, some points move around a little (mainly those with larger uncertainties, makes sense), but magnitudes, ‘patterns’ stay same. Repeating for 10. Only ~23,000 events now, down from ~41,000 at dec cut of –5. Similar effect to changing to 0. Points seem to mainly move ‘up’, I.e. become less negative or switch to positive.

Also now testing increasing energy bin from 10\*\*3.5 to 10\*\*5.5, to 10\*\*2 to 10\*\*5.5 at dec = -5. Now ~400,000 events, although higher fraction will now be atmospheric. Constraints are much worse, by ~3 orders of magnitude.

When testing at home couple days ago, or on David’s laptop on Monday, blue ‘global constraint’ bars did not show up. Should bring laptop in tomorrow and verify in morning that they work, as this is quite important part of plot (and presentation).

Ran through presentation again. Little quicker as expected. Noticed legend on plot with first 6 bins still had labels for DELS and QSO, so removed these. Have pushed latest slides to git to show supervisors at 2:30 (10 minutes time), although doubtful that blue bar will work.

Feedback:

* Slide at start giving aims/background. ‘don’t know mechanisms of production of these high energy neutrinos, want to find out’.
* Say energy bounds of bins
* Slide before overdensity map showing effective area (Felipe sent arxiv link to plot could use, could also use own plot, or both!)
* Could put atmosphere neutrinos slide in near start
* Be optimistic – don't dwell on no detection, say it is exciting that we can put upper bound on neutrino emissivity, which people can then use in future models
* S(epsilon) is spectral energy distribution
* Explain order one redshift dependent function is from spectral energy distribution and possibility of emissivity changing with time
* Say results are then order of magnitude constraints
* Try to save plots as png to get blue bar (and inset axes lines!)
* Spend longer with final plot, say why it is interesting to get to higher redshift (SFR peaks at z=2 so want to get to here).
* Can also add slide showing redshift distribution of two higher redshift surveys.
* Only took 10 minutes, have 20 in total, so don’t need to worry about time.

**Bold** = to do, normal = done

Added all ‘active points’ above. Rest just to do with what I say about slides. Z dist of QSO looked initially weird as no galaxies above/below z=1.5 for bin 0/1, so lines just stopped in midair. Put in fake points at dNdz=0 to make it look less weird.

18/08/22:

Presentation basically finished now, just need to practise. Blue bar and inset axes lines now show after switching to png, although obviously worse resolution. Best I can do unfortunately, but should be fine. Still need to talk longer on big final plot. Should say:

* Point out different galaxy surveys in different colours, list their names
* Say (1 sigma of) global constraint is now thinner due to more datapoints
* Constraints at much higher redshifts now, placing OoM limits on evolution of neutrino emissivity over time

Presentation done.

Need to now make sure code/work can be picked up by others after I leave tomorrow, I.e. upload everything to git (and maybe clean it up a bit).

Made plot of neutrino constraints for number flux, energy flux, and with/without l=2, with inset axes and uploaded this to the overleaf document.

Plan to upload folder to git called ‘Final\_Work’ or something similar, put everything important in there. May contain duplicates with other files on there but will make sure everything is latest version.

Started by going through jupyter notebooks (that are in top level of InternshipProject dir) alphabetically, finished this. Then started compiling any cool/useful plots in ‘Cool\_Plots’, which I will place into ‘Final\_Work’. Very few plots, so ended up just putting plots in ‘Final\_Work’ directly. Added presentation slides. Now just folders left, again going through alphabetically. Done. That should be everything, will check tomorrow morning if anything is missing/needs cleaning up, then push to git.

19/08/22:

Cleaned up directory a bit, then pushed to git. (And pushed this notebook to the same folder).