BaseAnalyser:

Runs checks on one sample\_id – runs checks on sample, proposes fixes and saves them to store

Do we need force twice – already in store??? – currently in twice

Not sure where to define project checks, not sure how to access checks

Need get fix to work as well

AnalyseMany (change name)/ analyse\_all: don’t think this needs to be a class OR maybe it is a class that gets called in an analyse() function and this is what gets used for cli.

If it is a class then its iterates over base analyser for required sample

Does this for all samples in a project/ for several sample ids (>1)/ for all sample ids

This needs to link up with the CLI

Sample generator – only a finite number of samples

Needs to be able to generate them all if no options provided

CLI:

Dachar analyse -p PROJECT -s sample id

If sample id/ ids provided – passed straight to analyser

Or

Dachar analyse -p PROJECT -v rh

I.e. provide details to create a sample id – which would then be generated by sample generator

(not sure what would be provided – fields you want to be specific about in the sample – any not provided would be \*)

If fields e.g. project/ sample id not provided at cli then runs for all projects or all samples etc. The sample generator would have to generate samples based on rules and what is provided at cli. But it also needs to work by just providing arguments.

Sample generator then passes these to grand analyser

Grand analyse uses base analyser

Also made fix template. Not sure if this is useful or where it comes into the above

Keeps hold of fixes

Once collected all of these propose them all at same tiem

Python dictionary in analysis – class represents datastructure – dictionary. – put methods on it to make it easier to interact witb

Proposals class. – make proposals

Class for analysis records

Keep track of proposals and then proposal them all at the end

Analysis

Separate fix proposal which contains history

Analysis record – might have propose fixes method

Loop through check object and if any have proposed fixes – then extract fixes and fix\_proposal\_store().propose()

Keep code agnostic(?)

Checks per project in options

But a lot are common

Function/class to wrap this info

Call get check list

Common checks local object here

Dictionary of project specific chcks

Get checks returns join of those two lists

Locate method -good for getting with strings locate('dachar.fixes.coord\_fixes.SqueezeDimensionsFix')

Object that lists all to get full path

To do:

Get workflow of analysing one sample to work – when call analyse()

-Fix proposal class ? x

-get ds\_ids from sample id – log in analysis record 🡪 need to get this to work to get all ds\_ids for zostoga (look at old analyser) x

-check characterised x

-check if already analysed 🡪 do I need this? Has force in json store as well x

-gets list of checks to carry out – function in options (should be referring to name of check or fix?)x

{{-carries out checks

– how to access from name of check? – locate but where would this function go

– keeps a record of whether any fixes needed – list of fixes?

-proposes fixes }} need to figure out a better way to do this

-sends analysis record to analysisrecordstore – need to add to this

-print statement to say analysis carried out on sample id

To test – will have to create fake instances

* test
* will have to sort splitting for different projects
* Go back to fix details template – needed or needed but written differently?
* Sample builder – to build from cli arguments/ based on rules?
* Analysing more than one sample – link to cli

Analyse():

Creates sample ids

Pass to AnalyseMany

AnalyseMany uses BaseAnalyser

This carries out the analyse process

Main problems now: running the checks:

How to get the check to run

Want to keep track of something when a fix is suggested – what is this – needs to return a list of the ds\_ids and their related fix

Can then iterate over this