## Mapping

Match = an automated or manual finding of a link between terms in two sources. This match can be lexical, logical or a combination.

Map = a curated, accepted alignment between two sources – maps are made up of term or concept matches.

## General

Ability to import initial terms or concepts for matching. Ability to export the mappings that you generate.

Ability to map strings from an input file

Ability to normalize and tokenize strings such that a possible match is more easily identified (not restricted to EXACT match, word order is not important, etc)

Map from input file to NCI Full\_Syn at the very least. Most of our use cases are mappings to NCIt

It would be nice to be able to map to other terminologies, such as those in the NCI Meta

Map to NCI concept properties (such as CAS numbers, UNII codes) and specify output tool to include such properties

Ability to exclude mappings to retired concepts – especially if mapping to NCIt

Output file must have the same number of records and in the same order, even if term is duplicated or a null value is returned, to aid in validation of matches.

Ability to persist and “pause” mappings. To curate a portion of matching results, then come back later to continue. This could be storage of maps in progress or ability to upload a map in progress to continue work. Ability to work over multiple sessions. Ability to export a map in progress.

Ability to collaborate on maps. Ability to view other people’s maps and make suggestions or comments. Ability to expand or edit other people’s maps in my own space – i.e. copy their map and add terms or change matches that I think are not correct.

Ability to not publish partial maps – keeping them hidden until they are fully curated. Possibly have the “publication” part of the maps be totally separate from collaboration tool?

We need a tool to map between two data sources in support of Cancer commons.

I want to take patient treatment charts, extract terms (possibly coded with ICD or local code lists) and map to an established source like NCIt for use in data aggregation.

If an algorithm does not find a match we want the ability to manually suggest and document matches.

## UI

Tool should respect accepted matches when escalating algorithms and not overwrite what the tool user has marked as a good match.

Any preference for online vs command line, or both? Online tool would be better for updating and distributing. Current command line tools are simplistic and demos have already replicated their functionality.

Tool should allow accepting and rejecting of matches.

Tool should mark failed matches – terms where no match was found.

Tool should show source and target coding scheme names and the terms that are being matches. Codes if available.

User authentication for eventual use by people outside the firewall.

Ability to comment on matches

Ability to rank or rate matches. Exact, partial, etc

If multiple matches occur, some way to rank matches so that the correct match (if it exists) is first in the record.

Ability to categorize match type – synonymy, logical, “related to”, broader, narrower

I want to be able to access the tool from remote locations. Some of our editors are not located at NCI. Initial use would be inside firewall

## Architecture

If tool is broken into multiple services, there will need to be a shared persistence or document format to move from tool to tool.

Tool realistically will not necessarily access all the back end options when first released.

Interaction with MEME will need to be detailed. Will their tool be isolated on MEME server?

Tool might need to access various back end processes – such as MEME browser, triple store and LexEVS.

Tool should allow escalating matches for performance. Possibly do simple text match first, then escalate to lucene, then look at parents and children.

### Notes from Corey demo

Application only works in Chrome. Will need to be updated to work in FF and likely Edge/IE. Need to check systems to see if we still have a list of browsers we need to support.

Has a google+ login, but that is being retired. Will need to change to just Google? Or will single sign on still be supported by Google

Have two columns – code and description. Liz asked if we need the description. Need to dive deeper on whether requiring description is a problem. Can we have other columns in addition, such as Semantic Type?

The organization/sorting of the imported value sets is inscrutable. Should be either customizable or, at least, predictable

Filtering should not require a whole word

Can type in correlation (is-a) but it might be handly for this to be a drop-down.

Automap is not currently working. This is one of the features we would most like mayo to support. Also, the comparison to a loaded terminology is going to be very useful.

Can map one to many and many to one. All maps are from left to right. For these, each of the constituent parts appears on its own line.

Maps and edits can be edited and saved to multiple versions. Can also be tagged as a FINAL version when you are finished working.

Once Automap is working, we will need to have automapped matches flagged differently than the manually curated matches.

Standardize does allow you to compare to a loaded terminology to map a singly loaded value set. It would give a score on how well it mapped against the terminology. Liz asked if we would choose terminologies or all of them? Corey said it looked at all and then ranked them on the best match.

The automap is a lucene modified contains. I would think we need a certain measure of control over the types of automapping are being used. Perhaps present a drop down to select from a list matching types.

We would need a way of curating beyond clicking each line.

Per Lyubov, we want the ability to choose to map against either names or synonyms (or maybe both). Also want the ability to match on code.

Automapped matches could maybe have “goodness” ranks that we can filter on. Show me all 90% confidence matches so I can curate those. Possibly one term could have a 90% match against one term, 70% against another, and 3 others at 40%.

Ability to adjust views. Right now the blocks are big and easy to read but that might make things more difficult for a large map. Ability to select various presentations – maybe a grid view.

The demo map was 80 terms, but one of the smaller maps Liz uses is a couple of thousand. This raises the issue that we should be able to sort and filter the matches themselves. “Please hide the matches that I have already curated”

Liz brought up HGNC map. She says it would be easy enough to sort on the left side to pull up just the new concepts. But how will the HGNC side be sorted? Will she need to filter or browser through all 35k HGNC concepts to find matches.

Liz requested that correlation and notes information would be good information to appear in the export.

Lyubov asked if the source set could have additional passive columns that might not be used in the mapping but that appear as informational – such as definition information. That information should be available for export, if the user chooses.

Currently any additional info would need to be in the value set import. It would be good if we can have that pulled in from a chosen terminology.

Lyubov asked about user accounts and if users can work on their own. We showed that there are user accounts but we need to figure out how those work. Corey said there is some function that isn’t working in GitHub but needs to check to remember what that is.

Kim has a mapping prototype that he will be deploying to Dev. It is set up to work against either TS or LexEVS. We discussed reusing the tomcat container and port assigned to the reportwriter, since no one is using it.