Possible MessyBrainz roadmap

To move MessyBrainz forward for GSoC, I think it might be a good idea to take modest approach and build some infrastructure for matching and clustering. To prove that the system works, one example script should address the easiest use cases that might yield quite a lot of results.

I think it would be good to start with:

1. Build out the core code to make sure that it supports common functions: make/add to cluster, collapse cluster, add more data to existing records.
2. Build a framework to run clustering scripts. This framework should connect to the MsB and MB databases and prepare everything so that the following types of scripts can run:
    a. Data cleanup scripts: Artist MBID lookup script. (Right now we have a lot of recording MBIDs, but we do not have any artist MBIDs. A script should look up recordings and find their artist_credit MBIDs and to add those to MsB.) This script should be runnable periodically and examine all records without artist MBIDs.
    b. Clustering scripts: Find new clusters, collapse clusters; also runnable periodically
3. The core matching functions of the scripts above should also be callable when new listens are being recorded into MessyBrainz. Ideally each of the scripts above would provide nothing more than a few overridable functions, such as setup, match and shutdown.
4. The matching script should provide a dry-run feature where the script examines the database and gives information about matching progress. This way we can adjust the scripts and algorithms and see their impact without actually changing the database. At the end each script should give a report that outputs all useful information related to the process. This might include:
    a. How many listens were examined and the total.
    b. How many clusters collapsed.
    c. What matches were edges cases and display those for review (if a 75% match confidence is specified for a script, this feature should output all matches that lie between 70% and 80% confidence)