EEGLAB2Hadoop: getting started

This repo contains tools to get started processing EEGLAB files with Hadoop using Hadoop Streaming with Python based MapReduce executables. If you are running Hadoop on a shared cluster, you might consider using virtualenv.

Files of interest (I know, all of them are interesting):

- EEGLAB2Hadoop / helpers / eeglab2hadoop.py
  - Convert EEGLAB .set files into Hadoop readable SequenceFile (with automatic compression)
  - Convert Hadoop ouput files into a single Matlab .mat file
  - Called from command line (see submit_job.sh for usage example)
- EEGLAB2Hadoop / mapreduce / with_hdfs / **mapper.py**
  - Executable receives a single record of a SequenceFile (e. g. 1 channel or IC of a dataset)
  - Calls process.py on that channel/IC
  - Emits KV pairs: *Key* is the dataset name, *Value* is the result returned by process.py. All results from the same dataset set be received by the same Reduce task
- EEGLAB2Hadoop / mapreduce / with_hdfs / process.py
  - Computes overlapping ERPs using multiple regression for a single IC or channel
- EEGLAB2Hadoop / mapreduce / with_hdfs / reducer.py
  - Receives KV pairs from Map tasks
  - Orders and consolidates the results of each dataset in an array
  - Emits KV pairs: *Key* is the dataset name, *Value* is the consolidated results for a single dataset. This writes the results to HDFS with filename containing *Key*.
- EEGLAB2Hadoop / mapreduce / with_hdfs / submit_job.sh
  - Drives all of the previous files mentioned.
  - 3 options:
    - `# Option -t: use test sequencefiles (small)`
    - `# Option -r: recompile sequencefiles`
    - `# Option -c: copy reducer outputs to local and convert to MATLAB format`
  - To create SequenceFiles use –r
  - Best practice it to debug MapReduce job using a small intput; use –t
  - To automatically copy the Python formatted results back to local and convert to Matlab format, use –c
  - Makes sure all folders for inputs and outputs exist
  - Configures the Hadoop job and submits it (see bottom of script for Hadoop parameters).
  - The following variables **are specific to your job**:

    - `DIR_LOCAL=$MY_STOR/data/RSVP`
    - `DIR_HDFS=/user/$USER/RSVP`
    - `START_IDX=44`
    - `END_IDX=60`
    - `FILESTR=$MY_STOR/data/RSVP/exp?/realtime/exp?_continuous_with_ica`

    This tells it to look for datasets START_IDX, END_IDX and all integers in between. It will substitute the dataset integer for the ? in FILESTR. (Ex: */data/RSVP/exp44/realtime/exp44_continuous_with_ica*). You will probably have to modify this scheme to your purposes.

You should be calling submit_job.sh to take care of all tasks. On the other hand, you may find it necessary to modify the structure of your SequenceFile or change compression settings (I never did get that feature to work, so it is disabled right now).