

## Laboratory Worksheet, Friday, March 19.

As always, there are several projects to work our way through today.

**I. Putative Highly Expressed Genes (PHX).** Not reached last time. We have to discuss the results of your search for data for this project, and agree on how to store it for common use.

Storage should be in the worksite *Math 548 001* which you should all belong to automatically at <http://ctng.umm.umich.edu>. Go to this page and make sure that you can access this site. If you are not registered for the 548 section, let me know. I will have to “authorize” you as a user. You will find here under *Resources* (a link from the worksite home page) a set of empty folders. We have to review responsibility for parts of this job, and the dimension of the data sets we will collect. There will next be folders for gathering the code you have been writing about component parts of the project.

**II. Training an HMM.** [Mainly a reprint.] This was mainly not finished last time. This will use the *new* command *dhmmem.m* in the *HMM* toolbox. There was a problem with this command because it contains an external, uncompiled C-command which wasn’t contained in the earlier version (*learn\_dhmm.m*). There are two possible fixes. Either edit the command file *dhmmem.m* where it invokes “normalize” with a C-suffix, and convert it to the corresponding *normalize.m* from KPMtools. Or you can download the old *learn\_dhmm.m* from the 548 web resources page. This does not involve the C-form of the *normalize* command.

I will only sketch the procedure here. You should first pull up the file for the command from the *HMM* directory (that is, the HMM toolbox, which should be at `/Desktop/KPM/HMM`) and read it. What are the inputs? What is the structure of the *data* input? We will have data for training a model where we will fix upon 2 hidden states beforehand, each with 6 emissions. Given this simple architecture of the model we are looking for, how many parameters are we looking to estimate in order to train the HMM model?

We will get the data by simply taking our original data sequence *O* and permuting it. Matlab has a random permutation generator called *RANDPERM* (you can look it up in the help window to see what its arguments are). The proposal is to permute *O* randomly about 50 or 60 times, and build an array of data sequences which can be used for Baum-Welch (EM) training.

Don’t bother prettying up the data report. We don’t have time.

### Appendix: Software Fixes.

**A. The Matlab Fix.** Go to the *548 Resources* link from the course home page. Download the file *KPM.tar.gz* onto your desktop. In the Mac’s these

should mount themselves on the desktop. Now call Matlab from the dock (it is the brown cone shaped icon in the bar at the bottom of the desktop). At the command line prompt (which looks a bit like >>) type `addpath ~/Desktop/KPM/HMM` [return], and so on, according to the names of the four downloads (.../KPM/KPMtools, KPMstats,netlab). This sets the paths to these functions for the permanently mounted Matlab in the lab. These mounted toolboxes will *disappear* when you log off. You can save them in your personal space, if you have a way to use them within Matlab there. I have not yet double checked the new `addpaths` since incorporating all the components for the HMM download into one local download.

*Please note that there is a Licence agreement which you are accepting for the Nabney netlab package.*

**B. The SeqAln Fix.** The local systems people have stored the Waterman program suite *SeqAln* locally at `http://www.math.lsa.umich.edu/courses/seqaln-2.0.dmg` as a disk image. Download this file to your desktop, where it becomes an icon. From the systems person here: “This is a MacOSX Jaguar binary, and the executable files can be found in that tree. So...students have to download and mount the dmg file and then type: `/Volumes/seqaln-2.0/seqaln/bin/SOMEPROGRAM` to run a program. Hope this helps.” This fix has been tested and works.