homework 10: the adventure of the moonlighting genes

Previous work from Irene Adler's laboratory has established that there seem to be a small number of gene batteries (modules of co-expressed genes) that are mixed and matched at different levels to specify the basic morphological properties of different sand mouse neuron cell types. These batteries involve about 100 different genes that her laboratory has identified. How many modules there are, and exactly which genes belong to which module, remain unknown.

Adler believes these 100 genes comprise three to six co-expressed gene batteries. She also believes that the batteries may share a few genes, and this overlap -- where the same gene is playing different functions in different contexts -- will be biologically informative.

non-negative matrix factorization & sand mouse neural cell types

The lab has collected RNA-seq data (as mapped read counts) for 60 different purified neuronal cell types. These data, as a simple whitespace-delimited table, are available here.

Adler has just read two papers, [Kim and Tidor 2003] and [Brunet et al 2004], that suggest that non-negative matrix factorization (NMF) is capable of identifying gene batteries, including shared genes between batteries.

You're visiting the Adler lab for a few weeks, sent by your PI Holmes in the hope of establishing a

collaborative relationship between the groups. She's not yet sure what to make of you, or of Holmes. She asks you if you can delve into the 1999 Lee and Seung Nature paper that popularized NMF and introduced an elegant mathematical algorithm to solve it. You say sure, you've taken MCB112; how hard could it be?

You set out to study the Lee and Seung paper, understand the derivation of their algorithm, implement NMF, and understand how it works -- and then, to analyze the Adler lab's data.

1. write a function that simulates positive control data

Using the generative model assumed by NMF, write a function that generates synthetic data for N genes and M experiments, generated from R underlying gene batteries.

Visualize your gene battery assignments to demonstrate that your function generates reasonable data.

2. implement nonnegative matrix factorization

Implement NMF, following the description in [Lee and Seung (1999)]. Explain your steps.

3. test your implementation

Apply your NMF function to synthetic datasets that you generate, varying the parameters of your synthetic data. What conclusions can you draw about how well NMF reconstructs the known gene batteries in your synthetic data? Explain your findings.

4. analyze the Adler data

Apply your NMF analysis to the Adler dataset.

- What is your best guess for how many gene batteries (NMF clusters) there are, and why? (Your choices are 3..6.)
- How many genes are in each battery, and what are they?
- How many moonlighting genes are there, and what are they?

turning in your work

Submit your Jupyter notebook page (a `.ipynb` file) to the course Canvas page under the Assignments tab. Please name your file `<LastName><FirstName>_<psetnumber>.ipynb`; for example, mine would be `EddySean_10.ipynb`.

Remember that we grade your psets as if they're lab reports. The quality of your text explanations of what you're doing (and why) are just as important as getting your code to work. We want to see you discuss both your analysis code, and your biological interpretations.

---

hints

- A "moonlighting" gene is a gene that has two very different functions in different contexts. Lee and Seung (1999) show an example of how non-negative matrix factorization of the word content of a large set of documents could separate different meanings (*polysemy*) of the word "lead" by putting it in different NMF components, one that groups "lead" with "glass", "copper", and "steel", and another that groups "lead" with "person", "example", "time", and "law". Brunet et al.

(2004) noted that this could also be useful in gene expression analysis, for detecting context-dependent patterns of gene expression -- i.e. seeing that gene X is sometimes co-expressed with one set of genes, and sometimes with another -- but unless I missed it in their paper, they don't show any good examples. Moonlighting genes would be an example of "polysemy" in gene expression analysis.

- In my version of the synthetic data generating script, I'm using N $\simeq$ 100, M $\simeq$ 60, R $\simeq$ 3-6, and $C \simeq 90000 - 110000$ per sample. I'm making gene batteries containing 10-40 genes each; and I'm putting $\simeq$ 2-5 moonlighting genes in two sets, but otherwise the gene batteries are disjoint. I say this to give you an idea of an appropriate scale for the synthetic data -- you'll also have to make some additional assumptions other than these.

- As you work through the derivation of the algorithm in [Lee and Seung (1999)], you'll see that there's a potential issue in the update equations they show in Figure 2, as we discussed in class, if your H are in units of probabilities, which is the natural thing to do if you're coming at this from the perspective of a generative model. Check the lecture notes. It's important to decide whether H is in units of counts (Lee and Seung style, which is more convenient from the perspective of matrix algebra notation) or in probabilities (my style, in which case you also want to keep track of total counts C_μ and expected counts $\lambda_{i\mu} = C_\mu \sum_a W_{ia} H_{a\mu}$), and stick to one way of doing it.

- The main point of parts 1 and 2 are the implementations themselves. It's worth getting a feel for how NMF works by exploring different synthetic data sets, but I

don't think there are really any crisp conclusions you can draw, so don't obsess too much about that. You'll develop some rough impressions, and that's what I'm looking for in terms of "conclusions".