

# Code Writeup

The code I've written in the project comes with a few folders, they are

- **Gaussian Block Model** - this is code from the second example in the Ramji/Montanari paper
- **LaTeX** - this just contains some of the documents I wrote for our meetings and Ramji's work he did for us.
- **Metagenomics factor** - this contains the code I made towards the Metagenomics problem
- **Point Mixture** - the code from the first example in the Ramji/Montanari paper
- **GOE** - Code that generates the Gaussian orthogonal ensemble used in both the Gaussian Block Model and the Point mixture.

The two examples from Ramji's paper are fairly self explanatory and can be easily derived again from the paper. So I'll just try and explain what the programs in **MetagenomicsFactor** do.

## Metagenomics Factor

The folder contains a number of MATLAB files as well as some folders, one folder contains the denoising non-linearities I used throughout the project and another contains files I used to test each method. The other programs are

- **ex\_basic/gene\_matrix** - is how I generated my examples, the gene\_matrix is the full model whilst the basic\_matrix is a simpler model generated by Gaussians.
- **gene\_factor** - The implementation from Ramji's paper that didn't work due to our columns not being orthogonal
- **gene\_testbed** - An older testing environment I used to plot results from the algorithm
- **gene\_U/V\_overlap** - how I would check my algorithm was doing well: this would take an estimate, apply the softmax nonlinearity (with inverse temp  $\mathbf{b}$ ) and find the frobenius overlap with the true matrix for  $U$  or  $V$
- **new\_lambda** When we first thought we needed to estimate our  $\Lambda$  signal to noise ratio we devised a way of updating our estimate, this is that method.

## Testing

The folder containing code I used to test parts of the algorithm may not be particularly interpretable as I would usually write the code up and then paste it into the command line when I needed it.

- `ieee_test` - was my first try at implementing the algorithm from the ieee paper (Phase Transitions and sample complexity in bayes optimal approximate message passing). I have a derivation of this written up in rough which I can write up properly if needed.
- `test_generic` - was my main base for testing the algorithm in Ramji's paper
- `test_initialisation` - This is how I tested different ways to start the algorithm, would plot graphs of average overlap and mse for each intialisation.
- `test_temp` - When I wanted to test how aggressive the softmaxs in the algorithm in Ramji's paper should be I would used this and record how the inverse temperature should change through each iteration.

## nonlinearities

This folder contains the different nonlinearities I would use.

- `gene_threshold` a really simple threshold for  $V$ , really aggressive and not used much
- `gene_Usoft/doftmax` The softmax for  $U$ , `doftmax` is the jacobian of the softmax
- `gene_V_soft/doftmax` The softmax for  $V$ , `doftmax` is the jacobian of the softmax
- `ieee_denoise` The denoiser for  $V$  used in the ieee paper reference above \*denoted  $f_X(\Sigma, T)$  in the paper
- `mean/var_normal_denoise` The denoiser also used in the ieee paper, its just for a normal prior so I used it for simpler cases and when I wanted to approximate distributions that would be difficult to compute posterior expectations for.
- `U_MCMC` I wanted to make an importance sampler for  $U$  here from Ramji's paper but wasn't able to.