$NNSC_B$ tested on a synthetic dataset

We generated a synthetic dataset with overlapping clusters as follows (Matlab notation is used). (Note that very large additive noise was superposed to the original data: the standard deviation of the noise was 75% of the standard deviation of the original data.) Figures 1a and 1b below depict the original clusters with and without additive noise.

```
eps = 1e-4;
n_samples = 21
n_genes = 35;
n_clust = 4;

alpha = eps*abs(randn(n_samples, n_clust));
gamma = eps*abs(randn(n_clust, n_genes));

const=1;

alpha(1:9,1) = 2*(const+rand(1,9))';
alpha(7:15,2) = 4*(const+rand(1,9))';
alpha(13:21,3) = 3*(const+rand(1,9))';
alpha(14:19,4) = 5*(const+rand(1,6))';

gamma(1,1:15) = const+rand(1,15);
gamma(2,11:25) = const+rand(1,15);
gamma(3,21:35) = const+rand(1,15);
gamma(4,3:12) = const+rand(1,10);

X = alpha * gamma;

noise_factor = 0.75;
X_orig = X;
X = X + abs(randn(size(X)))*noise_factor*std(X(find(X)));
```

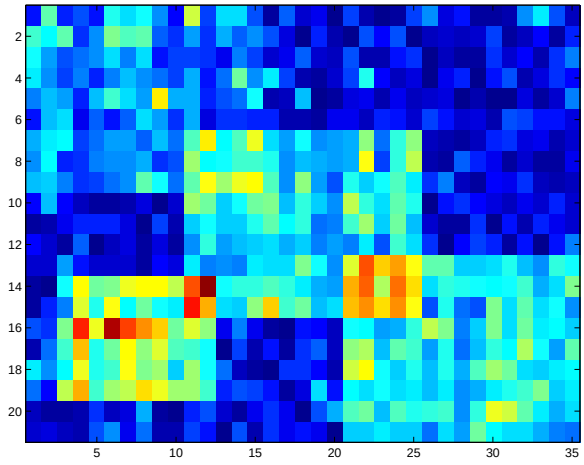
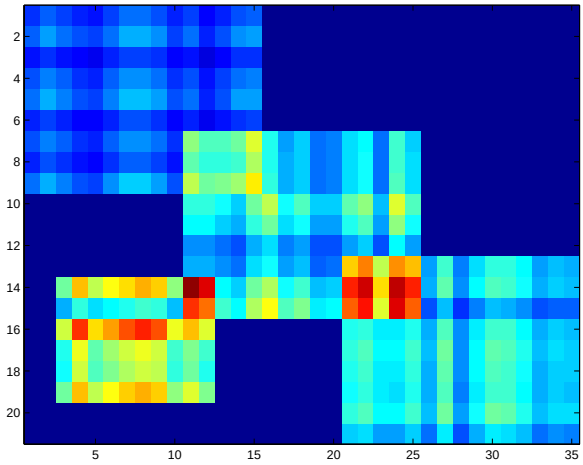


Figure 1a. Data (with noise)



1b. Data (without noise)

We ran NNSCB with increasingly larger λ (ranging from 0 to 0.75) and the true clusters as background knowledge (stepsize $\mu=10^{-5}$):

```
B(1, 1:15) = 1;
B(2, 11:25) = 1;
B(3, 21:35) = 1;
B(4, 3:12) = 1;
```

The following Table shows the relative error computed w.r.t. the noisy data X (`norm(X-A*S,'fro')/norm(X,'fro')`), as well as to the original data X_orig (an additive factor being added to compensate for the mean of the noise: `norm(X_orig - A*S+mean(noise),'fro')/norm(X_orig+mean(noise),'fro')`).

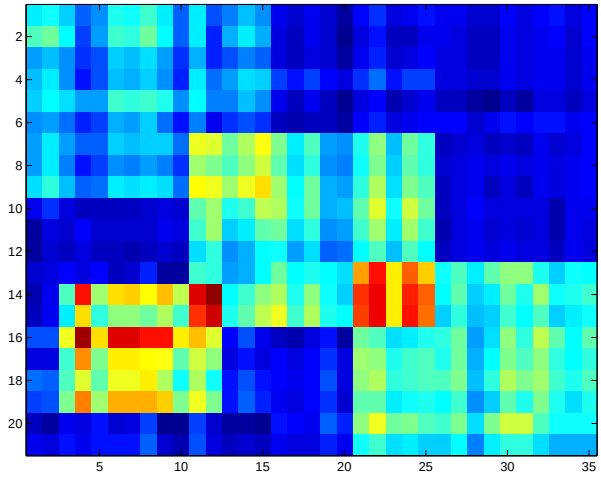
λ	0	0.1	0.2	0.4	0.5	0.75
relative error w.r.t. X	0.2069	0.2112	0.2186	0.2368	0.2464	0.2770
relative error w.r.t. X_orig	0.1292	**0.1224**	0.1248	0.1419	0.1532	0.1926

Using the threshold $1/\sqrt{n\_genes}$, we obtain the following 'gene' clusters:

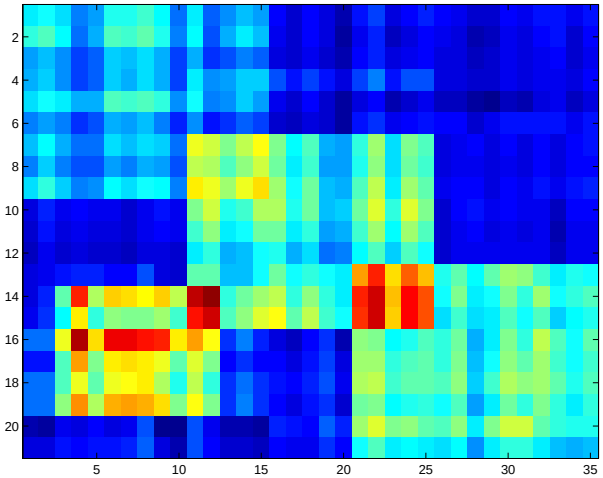
Original	λ=0	0.1	0.2	0.4	0.5	0.75
S (genes)						
3:12	3:12,32	3:12,32	3:12	3:12,32	3:12,32	4:12,21:22,26:27,29:33,35
1:15	1:3,5:9,11,13:15	1:3,5:11,13:15	1:11,13:15	1:15	1:15	1:15
21:35	21:35	21:35	21:35	21:35	21:35	21:35
11:25	11:18,20:25	11:18,20:25	11:18,20:25	11:18,20:25	11:18,21:25	11:18,21:25
A (samples)						
14:19	14:19	14:19	14:19	14:19	14:19	14:21
1:9	1:9,16,18:19	1:9,16:19	1:9,16:19	1:9,16:19	1:9,16:19	1:6,16:19
13:21	13:21	6,13,15:21	13:21	13:21	13:21	13,17:21
7:15	4,7:15	4,7:15	4,7:15	4,7:15	4,7:15	4,7:15

Note that the gene clusters are recovered almost perfectly despite the very large noise (75%), which we may encounter in real-world microarray data and that small values of λ tend to improve the clusters. (The missing elements are also improved, but some are just below the threshold used.) When 'extracting' sample clusters, we kept only the very largest coefficients of A. However, since the method is a gene clustering approach, we cannot expect the sample clusters to be recovered with high fidelity.

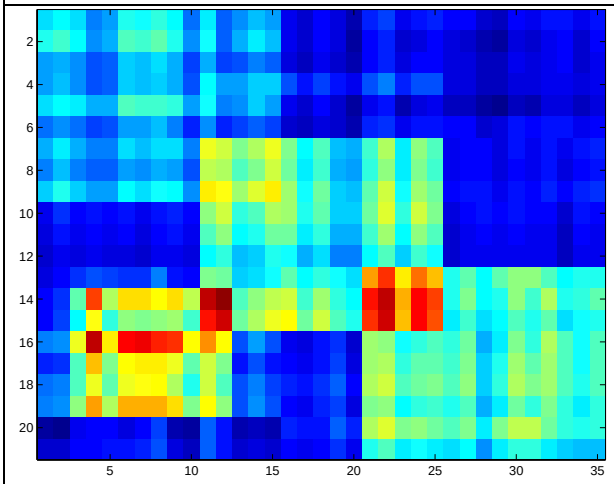
A*S



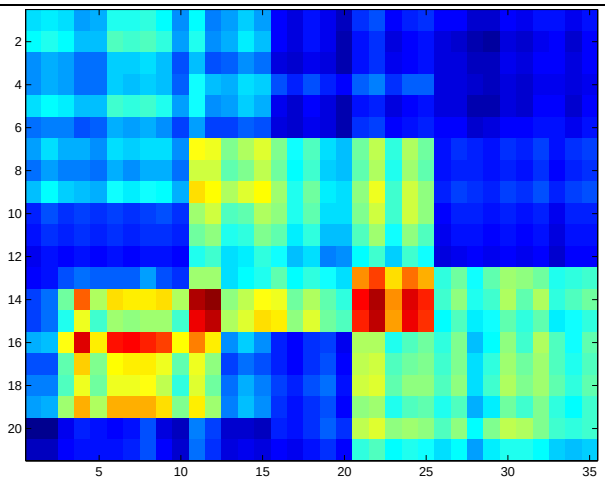
λ=0



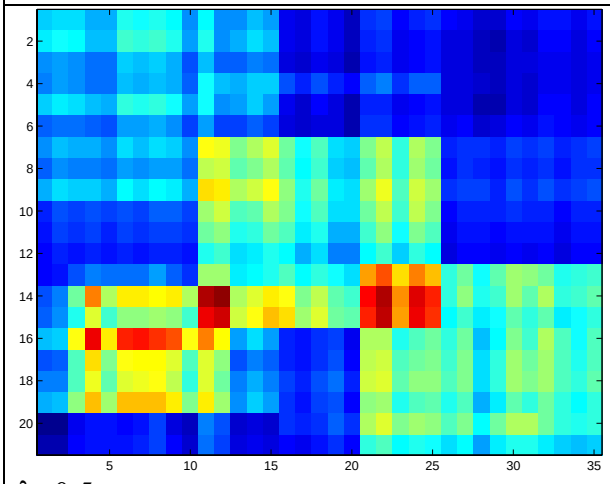
λ=0.1



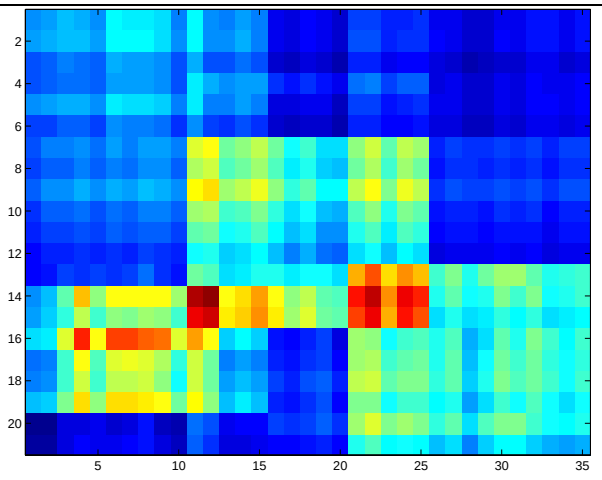
λ=0.2



λ=0.4



λ=0.5



λ=0.75