Implement a variant of JSNMF, JSNMF(same H) on mouse kidney dataset

Load data

```
addpath(genpath('../data'))
addpath('../codes')
addpath('../external')

load('kidney_5k_10k.mat')
load('housekeeping_genes.mat')
load('Kidney_sciCAR_data.mat','RNA')

% preparing for computing RAGI using ind1 ind2 data and so on
[~,ix1,ix2]=intersect(share_bd,RNA.Cells,'stable');
genes = RNA.Features; data = RNA.data;
data = data(:,ix2);
sM = sum(data,2); zero_row = find(sM==0);
data(zero_row,:) = []; genes(zero_row) = [];
data = data./repmat(sum(data),size(data,1),1)*10000;
data = log(data+1);

label_name = label;
tag = unique(label_name); tag = cellstr(tag);
[~, label] = ismember(label_name,tag);

% marker genes are given by the original publication;
% house-keeping genes are given in the website "http://www.housekeeping.unicamp.br/"
marker_genes = ["Slc12a3";'Trpm6';'Abca13';'Klhl3';'Wnk1';'Tsc22d1';'Cadps2';...
    'Egfem1';'Calb1';'Kl';'Temem72';'Dach1';'Ptprm';'Sgms2';'Tox3';'Frmpd4';...
    'Rbms3';'Fxyd4';'Dnm3';'Cacnb2';'Pde1c';'Pde8b'];
[~,~,ind1] = intersect(marker_genes,genes,'stable');
[~,~,ind2] = intersect(hp_genes,genes,'stable');
data = sparse(data);
clear ix1 ix2 sM zero_row tag share_bd RNA peaks marker_genes hp_genes genes

% parameter seletion for JSNMF(same H)
[gamma,Inits] = parameter_select_for_same_H(X1,X2,label);

% start time
tic
[~,~,H,iter,objs] = sameH_jsnmf(X1,X2,gamma,Inits);
```

```
number of iteration:10  obj:3784910.5566
number of iteration:20  obj:3761214.509
number of iteration:30  obj:3751801.21
number of iteration:40  obj:3746663.9154
number of iteration:50  obj:3743430.4192
number of iteration:60  obj:3741204.7831
number of iteration:70  obj:3739579.4619
number of iteration:80  obj:3738342.8442
number of iteration:90  obj:3737372.2481
number of iteration:100  obj:3736591.5778
number of iteration:110  obj:3735951.0168
number of iteration:120  obj:3735416.4074
number of iteration:130  obj:3734963.6767
converged!
```

```
disp('JSNMF_sameH runtime:');
```

```
JSNMF_sameH runtime:
```

```
toc
```

```
历时 104.327904 秒。
```

```
num_clu = length(unique(label));
A = cos_opt(H);  % cosine  %A = H*H'; %inner product
A = Wtrim(A,50);

% implement louvain clustering
[clust,~,~] = getNCluster(A,num_clu,0,3,20);
if length(unique(clust))== num_clu
   [ac, nmi_value, ~] = CalcMetrics(label, clust);
   [ave_mk_gini, ave_hk_gini, difgini] = RAGI(data,ind1,ind2,clust);
else
   [~, clust, ~] = SpectralClustering(A, num_clu);
   [ac, nmi_value, ~] = CalcMetrics(label, clust);
   [ave_mk_gini, ave_hk_gini, difgini] = RAGI(data,ind1,ind2,clust);
end
```

```
ac: 0.4435 2645/4753 nmi:0.4527
ave_marker_gene: 0.5772 ave_hk_gini:0.1715 diff_gini:0.4058
```

```
clear A S Inits alpha gamma clust iter
```