Decoding context

In the file DecodingDataRead.m, we first took the binarized calcium activity and took the sum of events in 2 s bins (60 frames). We then restricted analysis to:

(1) times when the mouse was running, and

(2) when the mouse was in context "1" or "2," representing two different cue configurations in the maze.

Finally, only activity from place cells were used. The resulting pre-processed data was saved for each mouse in a subfolder under the name binned_activity_decode.mat.

```
rootdir = "";
folders = ["M119"+filesep "M120"+filesep "M292"+filesep "M319"+filesep ...
    "M231"+filesep "M314"+filesep "M316"+filesep "M318"+filesep "M210"+filesep];
cohort = ["D1","D1","D1","D1","G1","G1","G1","G1","D1"];
sessions = ["early", "trained", "grouping"];
```

Estimate SVM performance for each animal & each condition.

```
ablate_percent=1;
ablation = 'None';

test_accuracy = nan(length(folders),3);
validation_accuracy = nan(length(folders),3);
s_max = length(sessions);

weights = cell(length(folders),3);
corrs = cell(length(folders),3);

parfor f = 1:length(folders)
    for s = 1:s_max
       try
            A=load(folders(f)+"binned_activity_decode_"+sessions(s)
+"_runrest_nopc.mat");

            ds = 20; % how many samples to downsample

            N_cells = size(A.a_use,2);
            N_cells_use = round(ablate_percent*N_cells);
            idx_cells_use = [];
            if strcmp(ablation,'random')
                % shuffle the cells
                subset_cells = randperm(N_cells);
                idx_cells_use = subset_cells(1:N_cells_use);
                corrs{f,s} = A.corr_coef(idx_cells_use);

            elseif strcmp(ablation, 'ascend')
                % sort the cells by increasing corr coef (most remap are
                % first)
                [corr_temp,idx_cells_use] =
topkrows(A.corr_coef,N_cells_use,'ascend');
                corrs{f,s} = corr_temp;

            elseif strcmp(ablation,'descend')
                [corr_temp,idx_cells_use] =
topkrows(A.corr_coef,N_cells_use,'descend');
                corrs{f,s} = corr_temp;
            elseif strcmp(ablation,'None')
                idx_cells_use = 1:N_cells_use;
                corrs{f,s} = A.corr_coef(idx_cells_use);
            end

            a_use_sub = A.a_use(1:ds:end,idx_cells_use);
            context_use_sub = A.context_use(1:ds:end);

            % % reshuffle context
            % shuffle_context_ind = randperm(size(context_use_sub,1));
            % context_use_sub = context_use_sub(shuffle_context_ind);

            hpartition =
cvpartition(context_use_sub,'Holdout',0.25,'Stratify',true);

            X_train = a_use_sub(training(hpartition),:);
            Y_trian = context_use_sub(training(hpartition));

            classificationSVM = fitcsvm(...
                X_train, ...
                Y_trian, ...
                'KernelFunction', 'rbf', ...
                'PolynomialOrder', [], ...
                'KernelScale', 'auto', ...
                'BoxConstraint', 1, ... % usuaally 1
                'Standardize', true);
                %'ScoreTransform','sign');

            % k-fold partitioning validation
            partitionedModel = crossval(classificationSVM, 'KFold', 20);
            [validationPredictions, validationScores] =
kfoldPredict(partitionedModel);
            validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');

            validation_accuracy(f,s) = validationAccuracy;

            % test set on hold out
            X_test = a_use_sub(test(hpartition),:);
            Y_test = context_use_sub(test(hpartition));

            [testPredictions,testScores] = predict(classificationSVM,X_test);
            testAccuracy = 1 - loss(classificationSVM,X_test,Y_test);

            rocObj_test = rocmetrics(Y_test, testScores,[1,2]);

            test_accuracy(f,s) = testAccuracy;

            % comparing SVM weights to remapping metric
            weights{f,s} = classificationSVM.Beta;

            display("Completed "+ folders(f)+"binned_activity_decode_"+sessions(s)
+".mat")

       catch e
            display("Skipping "+ folders(f)+"binned_activity_decode_"+sessions(s)
+".mat")
       end
    end
end
```

```
"Skipping M316\binned_activity_decode_early.mat"

"Skipping M231\binned_activity_decode_early.mat"

"Skipping M318\binned_activity_decode_early.mat"

"Skipping M319\binned_activity_decode_early.mat"

"Skipping M292\binned_activity_decode_early.mat"

"Skipping M314\binned_activity_decode_early.mat"

"Skipping M119\binned_activity_decode_early.mat"

"Skipping M120\binned_activity_decode_early.mat"

"Completed M292\binned_activity_decode_trained.mat"

"Completed M120\binned_activity_decode_trained.mat"

"Completed M292\binned_activity_decode_grouping.mat"

"Completed M231\binned_activity_decode_trained.mat"

"Completed M119\binned_activity_decode_trained.mat"

"Completed M319\binned_activity_decode_trained.mat"

"Completed M316\binned_activity_decode_trained.mat"

"Completed M120\binned_activity_decode_grouping.mat"

"Completed M119\binned_activity_decode_grouping.mat"

"Skipping M210\binned_activity_decode_early.mat"

"Completed M318\binned_activity_decode_trained.mat"

"Completed M231\binned_activity_decode_grouping.mat"
```

```
"Completed M316\binned_activity_decode_grouping.mat"

"Completed M210\binned_activity_decode_trained.mat"

"Skipping M210\binned_activity_decode_grouping.mat"

"Completed M314\binned_activity_decode_trained.mat"

"Completed M319\binned_activity_decode_grouping.mat"

"Completed M318\binned_activity_decode_grouping.mat"

"Completed M314\binned_activity_decode_grouping.mat"
```

Stats for Panel B

```
clear;
datat = readtable("decoder_results_deltas_repeat.xlsx");
WithinSubjectData =
table(categorical({'All','PC','NonPC','All','PC','NonPC'})',categorical({'D','D','D'
,'G','G','G'})',VariableNames=["CellSets","Tasks"]);

rm = fitrm(datat,'Score1-Score6 ~ 1','WithinDesign',WithinSubjectData);
ranova(rm,'WithinModel','CellSets+Tasks')
```

```
ans = 6×8 table
```

...

	SumSq	DF	MeanSq	F	pValue	pValueGG
1 (Intercept)	32.7610	1	32.7610	2.6045e+03	2.9046e-10	2.9046e-10
2 Error	0.0880	7	0.0126	1	0.5000	0.5000
3 (Intercept):CellSets	0.0785	2	0.0392	9.9173	0.0021	0.0125
4 Error(CellSets)	0.0554	14	0.0040	1	0.5000	0.5000
5 (Intercept):Tasks	0.0064	1	0.0064	7.4847	0.0291	0.0291
6 Error(Tasks)	0.0060	7	8.5972e-04	1	0.5000	0.5000

```
rm.multcompare('Tasks','by','CellSets')
```

```
ans = 6×8 table
```

...

	CellSets	Tasks_1	Tasks_2	Difference	StdErr	pValue	Lower
1	All	D	G	0.0198	0.0076	0.0343	0.0019
2	All	G	D	-0.0198	0.0076	0.0343	-0.0377
3	NonPC	D	G	0.0134	0.0106	0.2483	-0.0117
4	NonPC	G	D	-0.0134	0.0106	0.2483	-0.0385
5	PC	D	G	0.0363	0.0117	0.0170	0.0087
6	PC	G	D	-0.0363	0.0117	0.0170	-0.0638

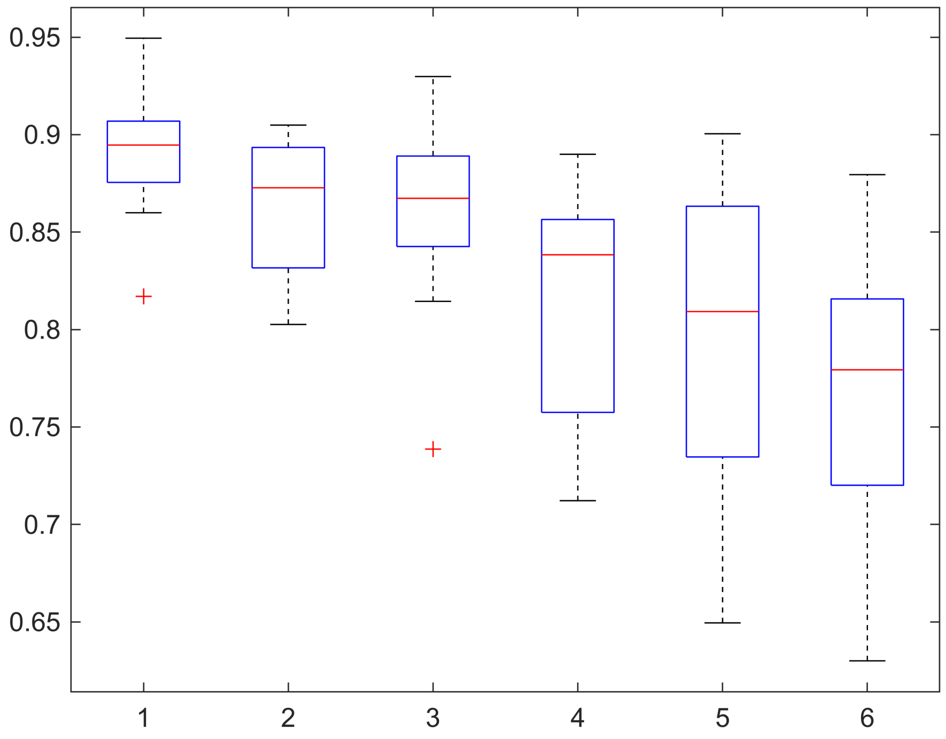
```
rm.multcompare('CellSets','by','Tasks')
```

```
ans = 12×8 table
```

...

	Tasks	CellSets_1	CellSets_2	Difference	StdErr	pValue	Lower
1	D	All	NonPC	0.1018	0.0251	0.0117	0.0280
2	D	All	PC	0.0331	0.0095	0.0247	0.0050
3	D	NonPC	All	-0.1018	0.0251	0.0117	-0.1757
4	D	NonPC	PC	-0.0688	0.0310	0.1356	-0.1601
5	D	PC	All	-0.0331	0.0095	0.0247	-0.0611
6	D	PC	NonPC	0.0688	0.0310	0.1356	-0.0225
7	G	All	NonPC	0.0954	0.0222	0.0087	0.0300
8	G	All	PC	0.0495	0.0111	0.0071	0.0169
9	G	NonPC	All	-0.0954	0.0222	0.0087	-0.1608
10	G	NonPC	PC	-0.0459	0.0284	0.3018	-0.1296
11	G	PC	All	-0.0495	0.0111	0.0071	-0.0821
12	G	PC	NonPC	0.0459	0.0284	0.3018	-0.0378

```
boxplot(table2array(datat(:,[3,6,4,7,5,8])))
```



```
load 'M314\binned_activity_decode_trained_all.mat'

[context_use_long,sort_i_long] = sort(context_use);
A_move_max = movmax(a_use(sort_i_long,:),[1000,1000],1);

disc_i = nansum(A_move_max(context_use_long ==
1,:),1)-nansum(A_move_max(context_use_long == 2,:),1);%./nansum(a_use,1);%
(nansum(a_use(context_use == 1,:),1)-nansum(a_use(context_use == 2,:),1))./
nansum(a_use,1);
disc_i = disc_i./nansum(A_move_max,1);

disc_i(isnan(disc_i)) = 0;
[~,sort_i] = sort(disc_i);

figure()
imagesc((1:size(a_use,1))/60,1:size(a_use,2),a_use(:,sort_i)')
yyaxis right
plot((1:size(a_use,1))/60,context_use,'w')
ylim([0.9,2.1])
```