Data structure:

D0: pooled control behavior for each experiment.
D1: inactivation behavior for each experiment.
DEL: difference between D1 and D0 for each experiment.

Dimensions of each 8D data matrix : {'subj', 'metric', loc, 'niter', 'k', 'nboot', 'exp', 'task'}

1) **subjects:** {'ctrl_mp', 'inj_mp'};
    i) ctrl_*: experiments with 3 sessions of no inactivation. inj_*: experiments with 3 sessions, one inactivation and two neighboring control sessions.
    ii) m/p/mp: manto/picasso/both.
2) **metrics:** { 'o1_dp', 'o2_dp'};
3) **locations**: {'all', 'ipsi', 'contra'};
4) **niter**: number of random trial splits
5) k: Number of trial splits:
    a) k=1: uses all trials, so niter=1 as well (no trial splits).
    b) k=2: uses 2 disjoint trial splits. niter=10 random trials plits.
6) **nboot**: number of bootstrap resamples from trial split.
7) **exp**: individual muscimol injection experiments.
8) **task**: tasks over which metric pattern is computed. Naturally, different metrics have different labels (e.g. O1 is 5 task labels, O2 is 6 labels).

Analysis to compute delta:

For given subject, metric and location (si, mi, li),
    -reshape into _trial resamples x exps x tasks_, which squeezes unused dimensions (k, niter).
    del_reshape = @(x, si, mi, li) reshape(nanmean(x(si,mi,li,:,:,:,:,:),5), size(x,4)*size(x,6), size(x,7), size(x,8));
    -compute mean error by first averaging over _tasks_ then over _exps_. This is because ntasks is not balanced across experiments (6 vs 10 tasks).
    del_mean = @(x) nanmean(nanmean(x,3),2);
    -resulting in a vector of delta estimates _trial_resamples x 1._ Mean and SD of this correspond to the estimate of global delta and the SE of this estimate.

Analysis to compute sparsity:

For a given experiment, compute sparseness of each bootstrap estimate of the delta pattern. Mean and SD of this correspond to the estimate of global sparsity and the SE of this estimate.

sparseness = @(x) (1-(nanmean(x).^2 ./ nanmean(x.^2))) ./ (1-length(x)^-1);