Core steps in search fit (assume 3 scans, 3 DCT terms per scan)

**data**: converted to percent signal (by scan)

**trendBetas**: pinv(**trends**)\***data**; (*9 trends, 9 betas* – DC betas are small)

**data**: detrend using **trendBetas**

**rawrss**: sum(double(**data**).^2) (raw ss on detrended data)

rf: Gaussian with height 1

pred: conv((im \* rf)^n, hrf)

b: pinv([pred **trends**])\* **data** (*4 trends, 4 betas)*

rss: norm(**data**-[pred **trends**]\*b).^2;

model.b: combine b and **trendBetas** (*10 betas)*

**rmSearchFit**

* l 273: rmLoadData (raw, then convert to PC *by scan*)

**data** = rmLoadData(view,p2,slice,coarse);

* l 152: Make trends

[**trends**, nTrends, dcid] = rmMakeTrends(params);

* l 283: fit trends

**trendBetas** = pinv(**trends**)\***data**;

* l 284: remove them

**data** = **data** - **trends**\***trendBetas**;

* l 306: Compute Raw SS

s{n}.**rawrss**(wProcess) = sum(double(**data**).^2);

* l 349: Extract DC terms from trends

t.**trends** = **trends**(:,dcid);

* l 429: Fit pRF model

**rmSearchFit\_oneGaussianNonlinear**

* + l 70: Get (x,y,sigma, n) returned
  + l 90: remake RF

rf = exp( (Yv.\*Yv + Xv.\*Xv) ./ (-2.\*(outParams(3).^2)) );

* + l 91: unconvolved prediction

pred = (params.analysis.allstimimages\_unconvolved \* rf).^ n;

* + l 94: convolved prediction

pred(inds) = filter(params.analysis.Hrf{scan}, 1, pred(inds));

* + l 97: Augment prediction with DC trends

X = [pred **trends**];

* + l 98: Beta weights (for prediction and DC trends)

b = pinv(X)\*vData;

* + l 99: Residual SS

rss = norm(vData-X\*b).^2;

* + l 113: Store RSS

model.rss(vi) = rss;

* + l 114: Store Beta weights (for prediction and DC trends)

model.b([1 t\_id],vi) = b;

* l 442: Add initially computed, prediction-free trend betas (trendBetas) to newly computed trend betas (derived together with model predictions)

s{mm}.b(nB-ntrends+1:end,:) = s{mm}.b(nB-ntrends+1:end,:)+**trendBetas**;

* l 457: Save

**rmMain**

**rmGridFit**

Make predictions

* l 113: Get the unconvolved images:

allstimimages=params.analysis.allstimimages\_unconvolved;

* l 119: Make the RFs:

rf = rfGaussian2d(params.analysis.X ...

* l 126: Unconvolved predictions:

pred = allstimimages\*rf;

* l 141: Exponentiate:

prediction = bsxfun(@power, prediction, params.analysis.exponent');

* l 145: Convolve with HRF:

prediction(inds,:) = filter(hrf, 1, prediction(inds,:));

* l 154: Decimate:

prediction = rmDecimate(prediction,...

Load Data

* l 169: rmLoadData

**rmLoadData**

* + l 64: LOOP OVER STIM
    - l 100: rmLoadDataROI ==>
    - l 160: loadTSeries (raw - whole brain)
    - l 164: Spatial Smoothing

dhkGraySmooth

* + - l 170: Restrict to ROI

tSeries = tSeries(:,roiIndex(coarseIndex));

* + - l 173: Convert to PC

tSeries = raw2pc(tSeries);

* + l 116: Put tSeries in data
  + l 117: END LOOP OVER STIM

Detrend

* l 186: make trends

[trends, ntrends, dcid] = rmMakeTrends(params);

* l 190: fit them

trendBetas = pinv(trends)\*data;

* l 195: remove them

data = data - trends\*trendBetas;

* l 203: decimate data

data = rmDecimate(data,params.analysis.coarseDecimate);

* l 204: decimate trends

trends = rmDecimate(trends,params.analysis.coarseDecimate);

* l 207: compute RSS,

rssdata = sum(data.^2);

* l 250: keep dc terms in trends for fitting pRF

t.trends = trends(:,dcid);

* l 347: Fit pRF model:

**rmGridFit\_oneGaussianNonlinear**

* + l 19: convert RSS from sum to mean

model.rss=single(model.rss./(size(prediction,1)-size(trends,2)+1));

* + l 29: GRID FIT. Loop over predictions
    - l 55: Augment prediction with **DC trends**

X = [prediction(:,n) trends];

* + - l 58: Get betas (for predictor and DC trends) and residual ss for all voxels, one prediction

[b,~,rss] = lscov(X,data);

* + - l 74: If the RSS is lower than the stored RSS, keep this solution

minRssIndex = rss < model.rss;

* + l 101: Convert RSS back to sum (rather than mean)

model.rss=single(model.rss.\*(size(prediction,1)-size(trends,2)+1));

* l 359: Add initially computed, prediction-free trend betas (trendBetas) to newly computed trend betas (derived together with model predictions)

s{mm}.b(nB-ntrends+1:end,:) = s{mm}.b(nB-ntrends+1:end,:)+trendBetas;

* l 373: Interpolate data (if coarse sampling)

model = rmInterpolate(view, model, params);

* l 379: Save

**rmSearchFit**

* l 113: Load grid fit model

load(rmFile);

* l 152: Make trends

[trends, nTrends, dcid] = rmMakeTrends(params);

* l 214: Get all coords

allcoords = viewGet(view,'coords', slice);

* l 220: Restrict to ROI

[tmp, wProcess] = intersectCols(allcoords,ROIcoords);

* l 251: Reset all betas except prediction betas to 0

s{n}.b(2:nTrends+1,wProcess) = 0;

* l 273: rmLoadData

**rmLoadData**

* + l 64: LOOP OVER STIM
    - l 68: loadTSeries (raw - whole brain)
    - l 96: Convert to PC (one scan at a time!!)

tSeries = raw2pc(tSeries);

* + l 116: Put tSeries in data
  + l 117: END LOOP OVER STIM
* l 283: fit trends

trendBetas = pinv(trends)\*data;

* l 284: remove them

data = data - trends\*trendBetas;

* l 306: Compute RSS

s{n}.rawrss(wProcess) = sum(double(data).^2);

* l 325: Get unconvolved stim images

params.analysis.allstimimages\_unconvolved = rmDecimate(...

params.analysis.allstimimages\_unconvolved, doDecimate);

* l 349: Extract DC terms from trends

t.trends = trends(:,dcid);

* l 429: Fit pRF model

**rmSearchFit\_oneGaussianNonlinear**

* + l 70: Fit pRF model

fmincon(@(x) rmModelSearchFit\_oneGaussianNonlinear(x,vData,...

**rmModelSearchFit\_oneGaussianNonlinear**

* + - l 20: remake Gaussian

RF = exp( (Yv.\*Yv + Xv.\*Xv) ./ (-2.\*(p(3).^2)) );

* + - l 23: Unconvolved prediction

pred = (stim\*RF).^p(4);

* + - l 27: Convolved prediction
    - pred(inds,:) = filter(hrf{scan}, 1, pred(inds,:));
    - l 30: Augment prediction with DC trends

X = [pred t];

* + - l 34-52: Get beta weight and resisual sum of squares using SVD
  + l 70: Get (x,y,sigma, n) returned
  + l 90: remake RF

rf = exp( (Yv.\*Yv + Xv.\*Xv) ./ (-2.\*(outParams(3).^2)) );

* + l 91: unconvolved prediction

pred = (params.analysis.allstimimages\_unconvolved \* rf).^ n;

* + l 94: convolved prediction

pred(inds) = filter(params.analysis.Hrf{scan}, 1, pred(inds));

* + l 97: Augment prediction with DC trends

X = [pred trends];

* + l 98: Beta weights (for prediction and DC trends)

b = pinv(X)\*vData;

* + l 99: Residual SS

rss = norm(vData-X\*b).^2;

* + l 113: Store RSS

model.rss(vi) = rss;

* + l 114: Store Beta weights (for prediction and DC trends)

model.b([1 t\_id],vi) = b;

* l 442: Add initially computed, prediction-free trend betas (trendBetas) to newly computed trend betas (derived together with model predictions)

s{mm}.b(nB-ntrends+1:end,:) = s{mm}.b(nB-ntrends+1:end,:)+trendBetas;

* l 457: Save