COSMO MVPA WORKSHOP

Ariana Familiar

May 21 2019

http://cosmomvpa.org/index.html

Outline

- MVPA review
- CoSMo concepts
- Demos
  - ROI-based LDA classification
  - Searchlight with Naive Bayes classifier
  - Representational Similarity Analysis
  - Monte Carlo permutation testing with TFCE
- Exercise

- Slides, code, & data available for download:
  - **Git:** https://github.com/afamiliar/cosmoMVPA-workshop.git
  - **Google Drive:** https://drive.google.com/drive/folders/1Evjz9ovrejl4UWYR0VV7oiLd7G7Gqn_o?usp=sharing

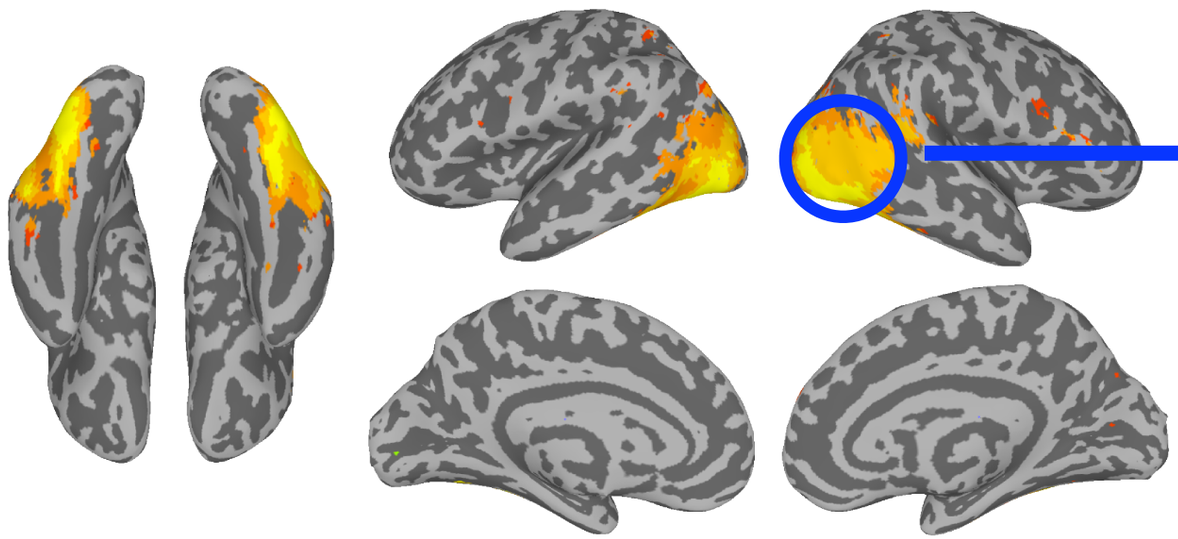
MVPA

- "Multi-variate pattern analysis"
- Unlike univariate measures which aggregate across data points, MVPA takes into account spatial *patterns* across a set of data points
    - sensitive to fine-grained spatial patterns that might discriminate between experimental conditions

MVPA

- Can be used for:
  - volumetric fMRI data
  - surface-based fMRI data
  - MEEG timelocked data
  - MEEG time-frequency data

MVPA



“pattern”

“features”

MVPA

- Typical MVPA:
  - Classifier-based / "pattern classification"
  - Representational similarity matching / "RSA"

MVPA

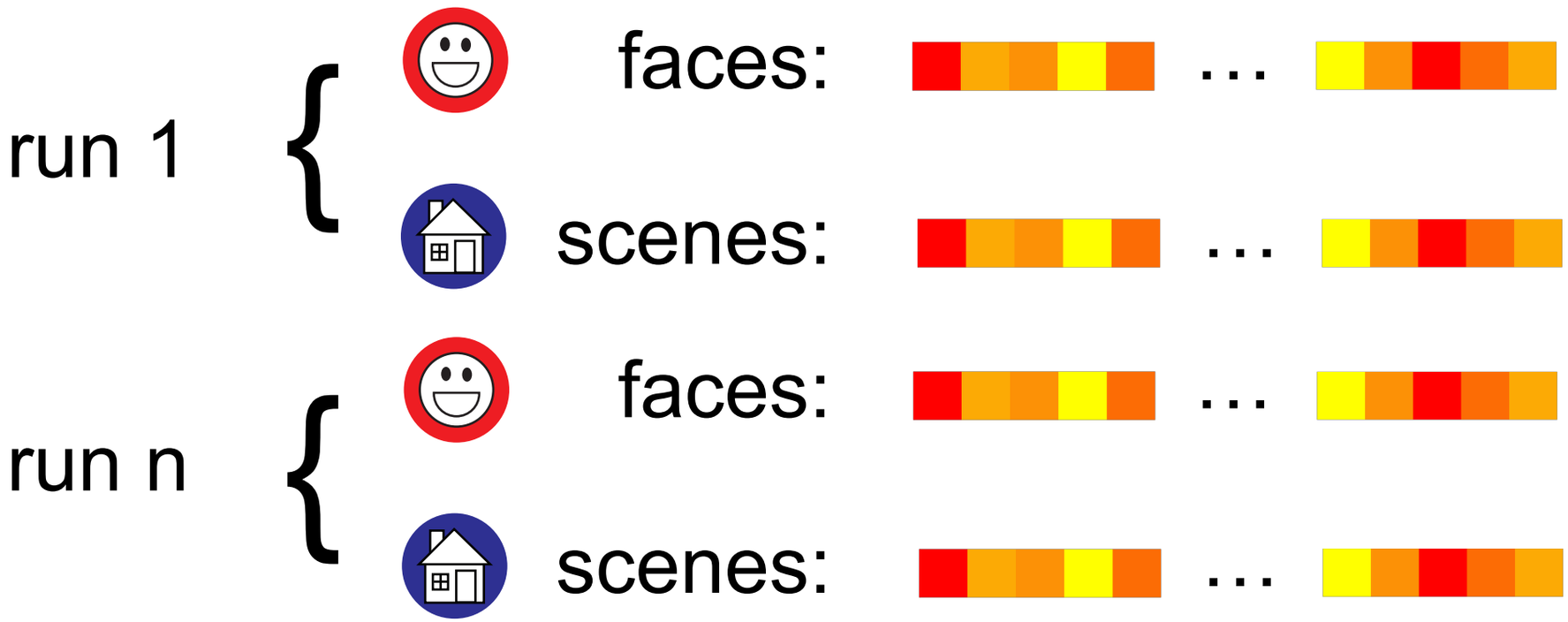
- Classifier-based MVPA

faces:  … 

scenes:  … 

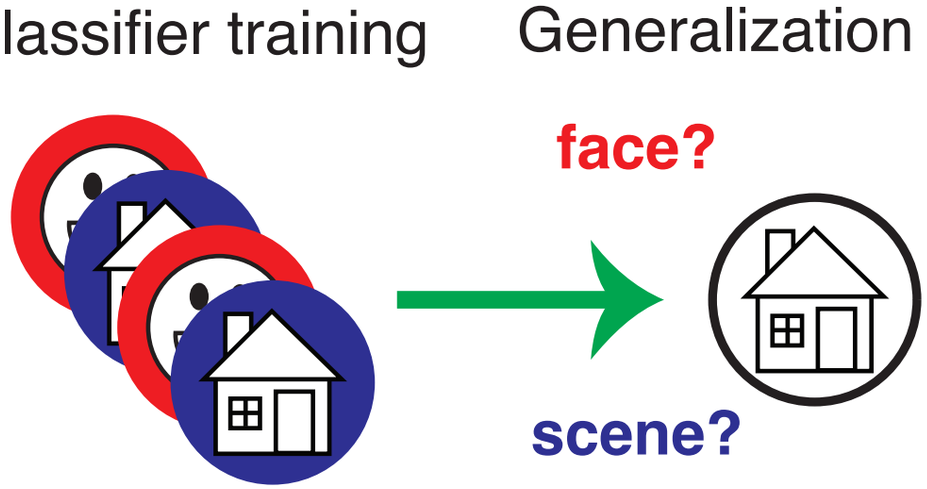
MVPA

- Classifier-based MVPA



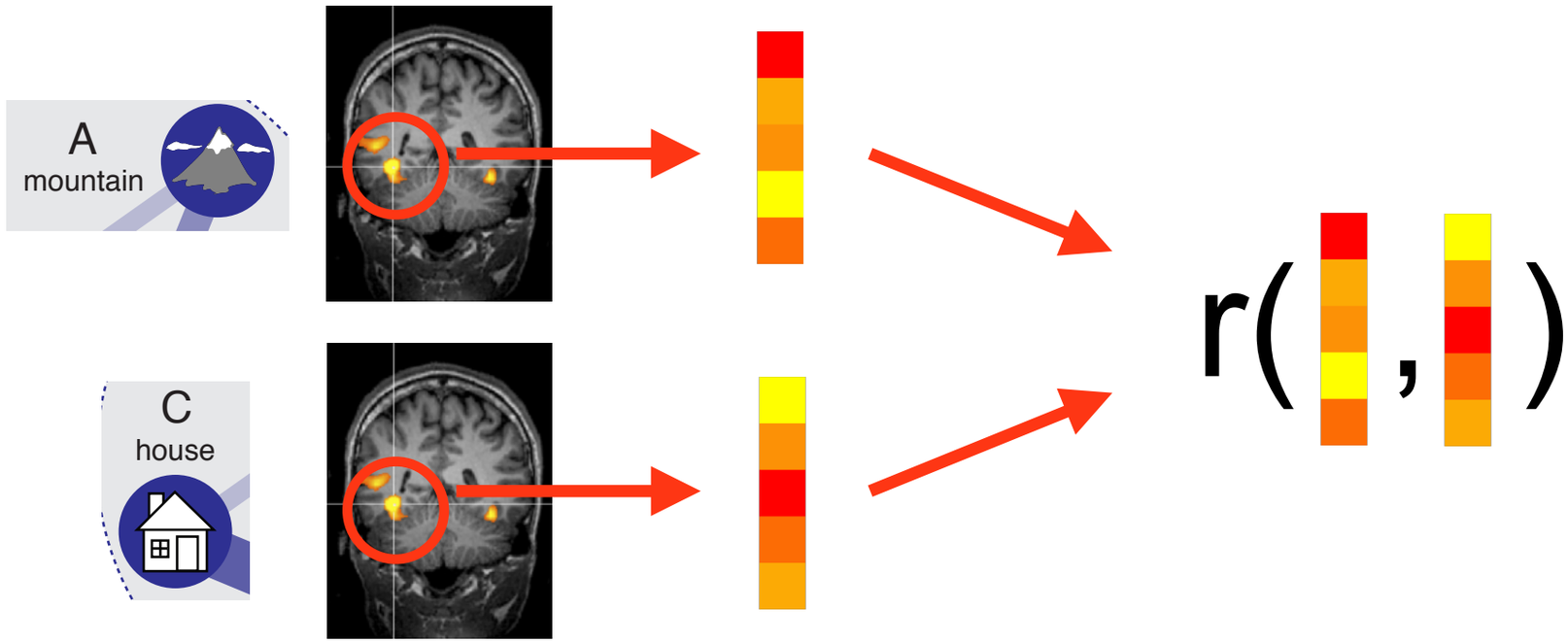
MVPA

- Classifier-based MVPA



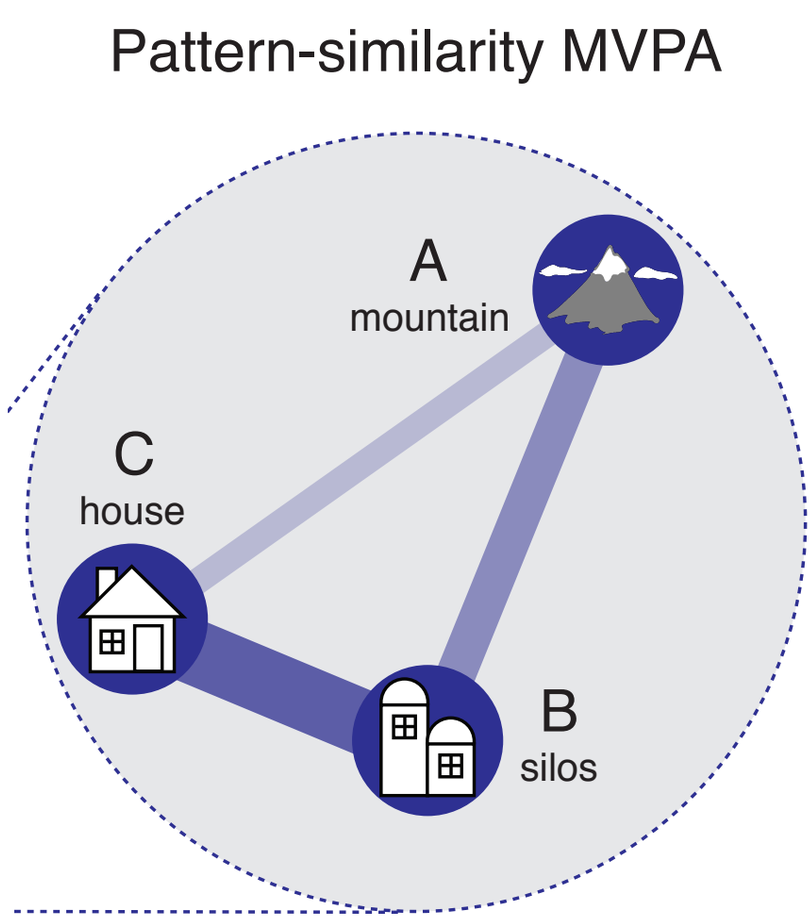
MVPA

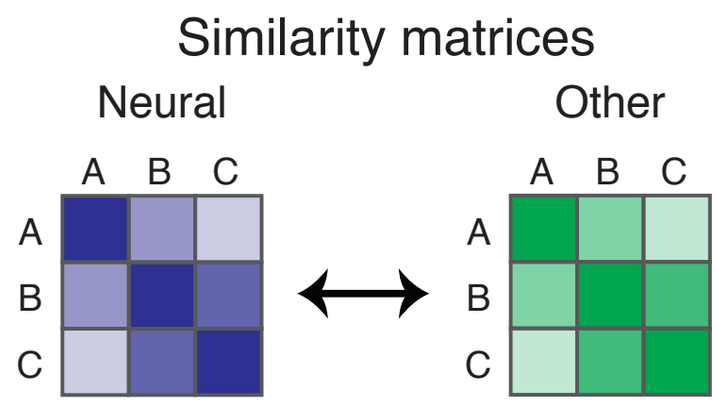
- Representational similarity analysis



MVPA

- Representational similarity analysis

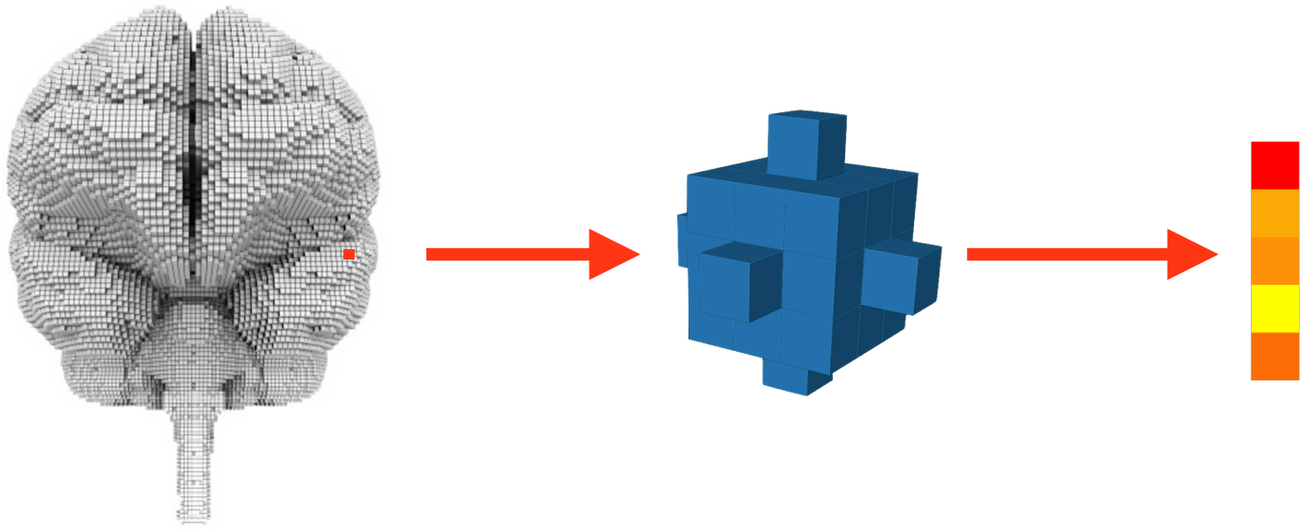




Compare matrix of pairwise similarities of multi-voxel patterns to matrix of pairwise similarities from another domain of interest

MVPA

- Whole-brain searchlight
    - Not defining any region a priori



MVPA

- Statistical testing
  - Multiple comparisons correction via permutation testing:
    - Re-sample data
    - Run same test
    - Do this many times to construct a ‘null’ distribution
    - Compare original value to null distribution

CoSMo MVPA

- free/open-source
- Runs on Matlab & GNU Octave platform
  - MS Windows, OSX, GNU/Linux
  - Can parallelize functions if run on GNU
- Supports a wide-range of data formats

http://cosmomvpa.org/index.html

CoSMo MVPA

- Why I like it:
  - Based in Matlab
  - Easy to use/compare different classifiers
  - Easy to do subject- & group-level analyses
  - Easy to use same code across different projects
  - Easy to do both MVPA and statistical tests in the same pipeline
  - Clear and thorough documentation & example code

CoSMo MVPA

- Getting started:
  - Matlab or Octave installation
  - Source code
    - Matlab programming experience
    - fMRI or MEEG data analysis experience
    - A basic understanding of MVPA concepts

http://cosmomvpa.org/index.html

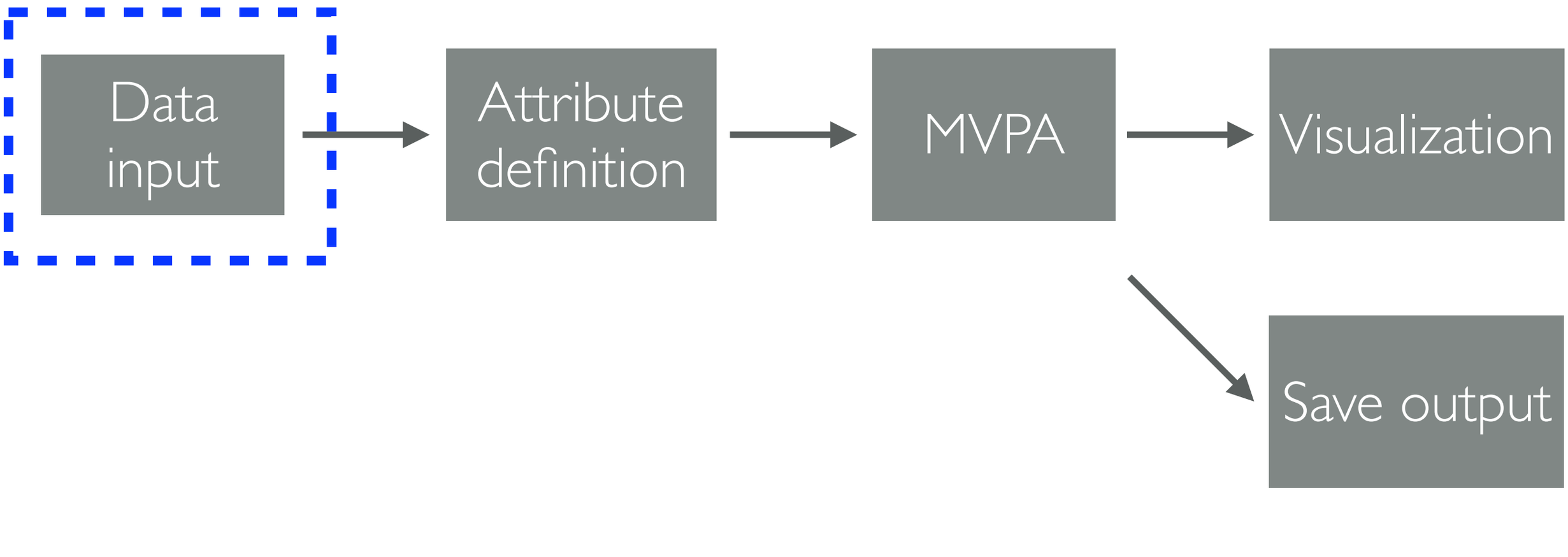
CoSMo MVPA

- Getting the source code, two options:
  - **git users:**
    - git clone https://github.com/CoSMoMVPA/CoSMoMVPA.git
    - make -C CoSMoMVPA install
  - **Download zip archive and use cosmo set path or Matlab's setpath(genpath('path to source code directory'))**
    - https://github.com/CoSMoMVPA/CoSMoMVPA/archive/master.zip
    - http://cosmomvpa.org/matlab/cosmo_set_path.html#cosmo-set-path

http://cosmomvpa.org/index.html

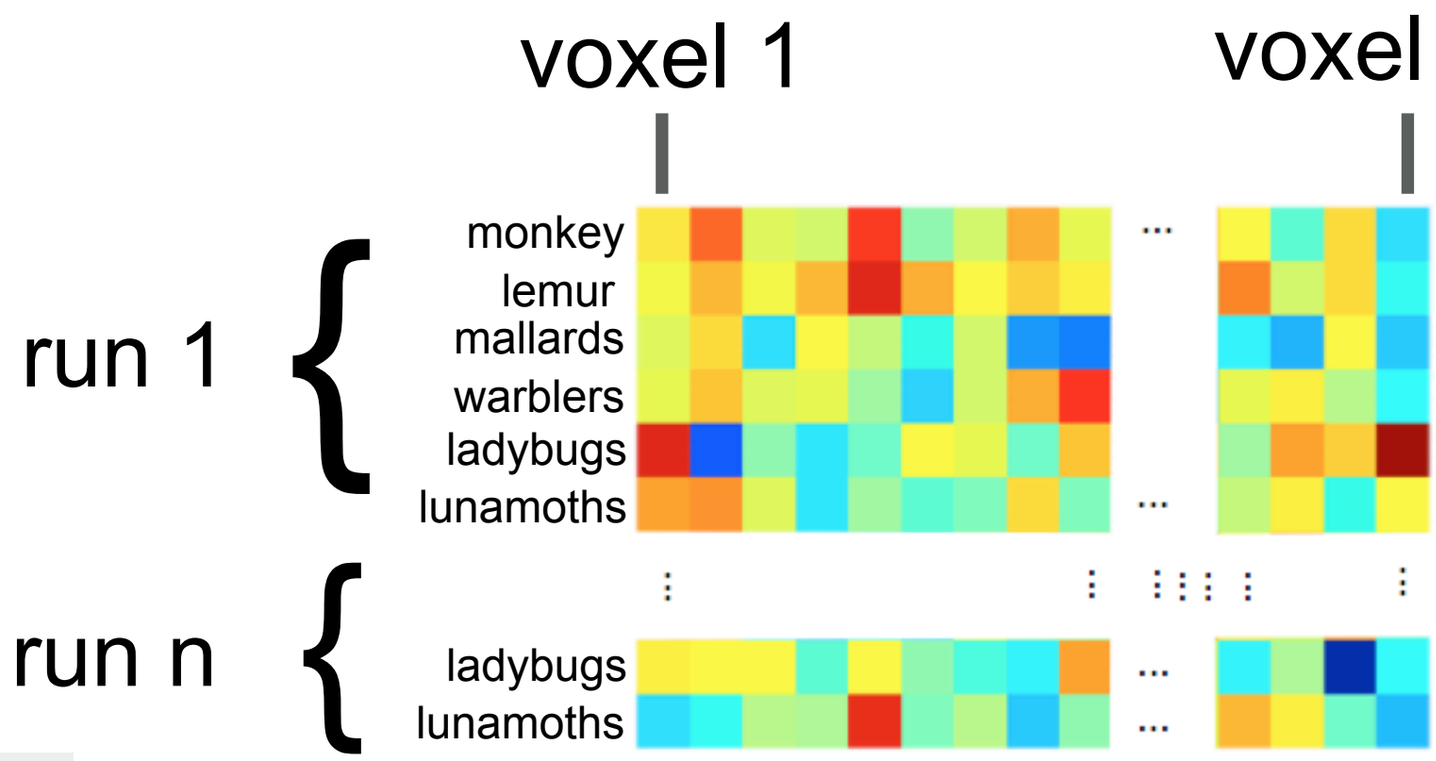
CoSMo MVPA

- Analysis pipeline (within subject)



CoSMo MVPA

- Data input
  - Input data structure

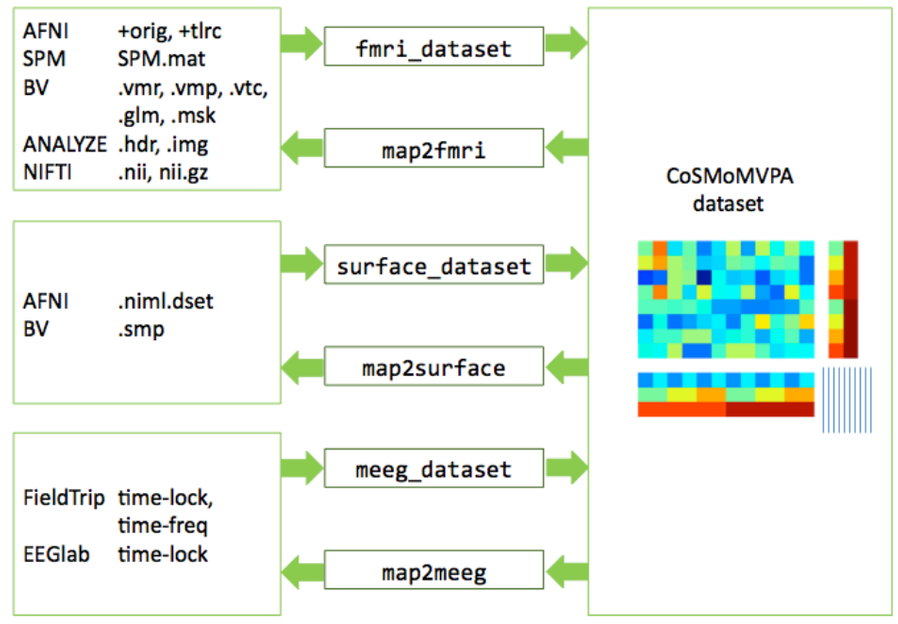


(This is for all brain voxels)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Data input
  - Can use either external toolboxes, or CoSMo conversion code to map to/from this dataset representation, e.g.:
    - cosmo_fmri_dataset
    - cosmo_surface_dataset
    - cosmo_meeg_dataset
    - cosmo_map2fmri
    - cosmo_map2surface
    - cosmo_map2meeg



http://cosmomvpa.org/index.html

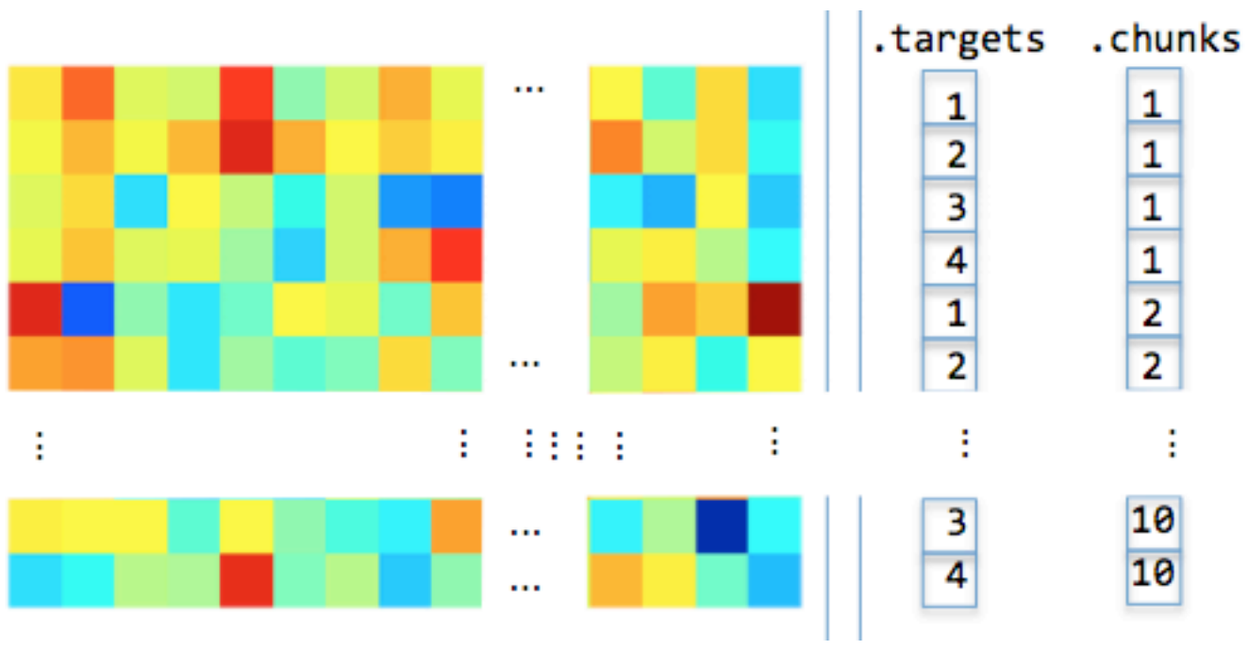
CoSMo MVPA

- Data input
  - Input mask (binary, whole-brain)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Data input
  - Targets: stimuli/condition ID numbers
  - Chunks: run ID numbers



http://cosmomvpa.org/index.html

CoSMo MVPA

- Data input
  - Example code:

```
num_cond = 6;
num_runs = 10;
targets  = repmat((1:num_cond)',num_runs,1);
chunks   = floor(((1:num_runs*num_cond)-1)/num_cond)'+1;

dataset  = cosmo_fmri_dataset('glm_tstats.nii', ...
                              'mask', 'visual_cortex_mask.nii', ...
                              'targets', targets,...
                              'chunks', chunks);
```

http://cosmomvpa.org/index.html

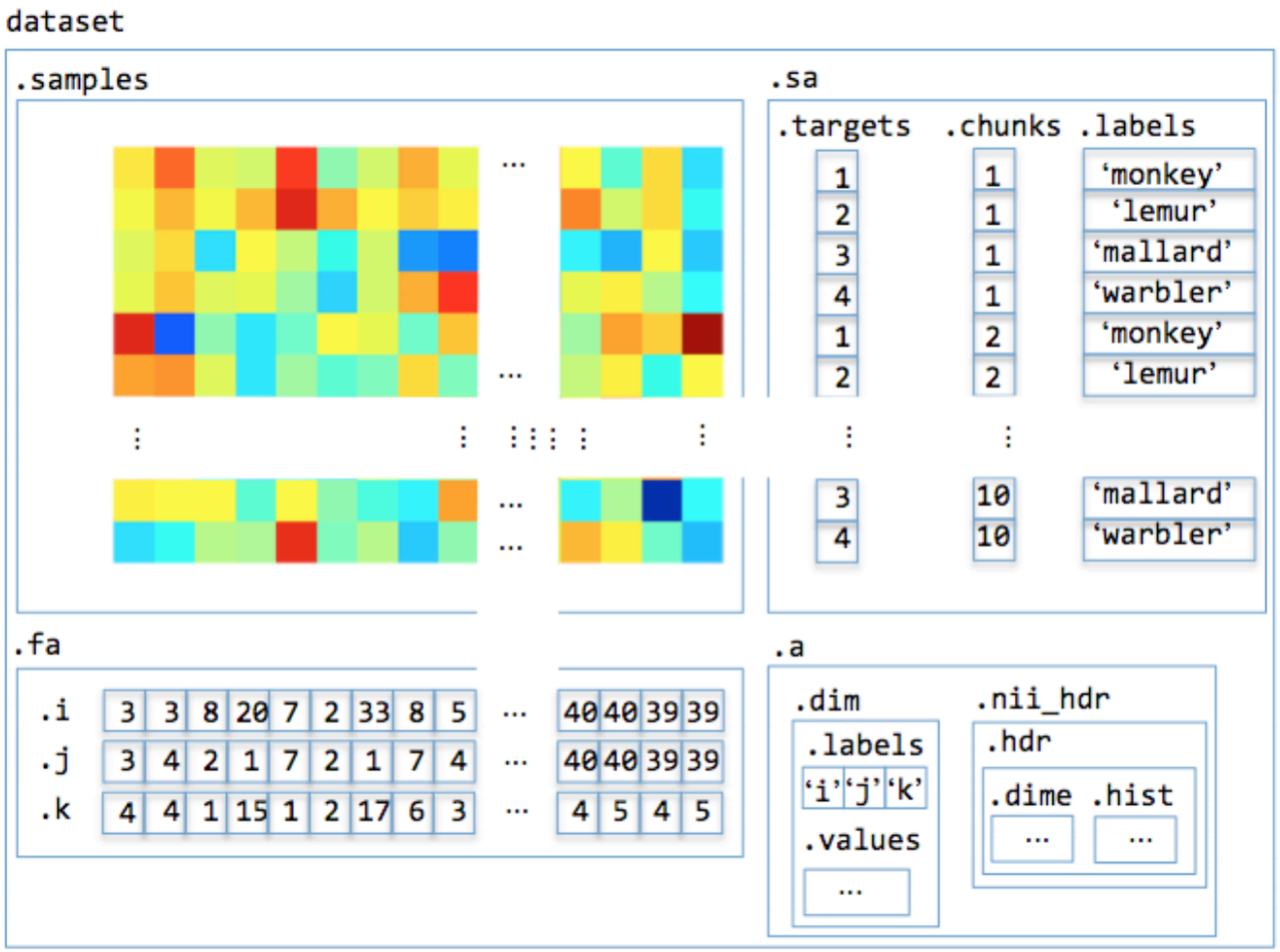
CoSMo MVPA

- Data input
  - samples: data (dimensions = num_stim*num_runs x num voxels in mask)
  - fa: feature attributes (e.g. voxel location)
  - sa: sample attributes (e.g. condition or run number)
  - a: additional data set attributes, such as dimensions of volumes and voxels

http://cosmomvpa.org/index.html

CoSMo MVPA

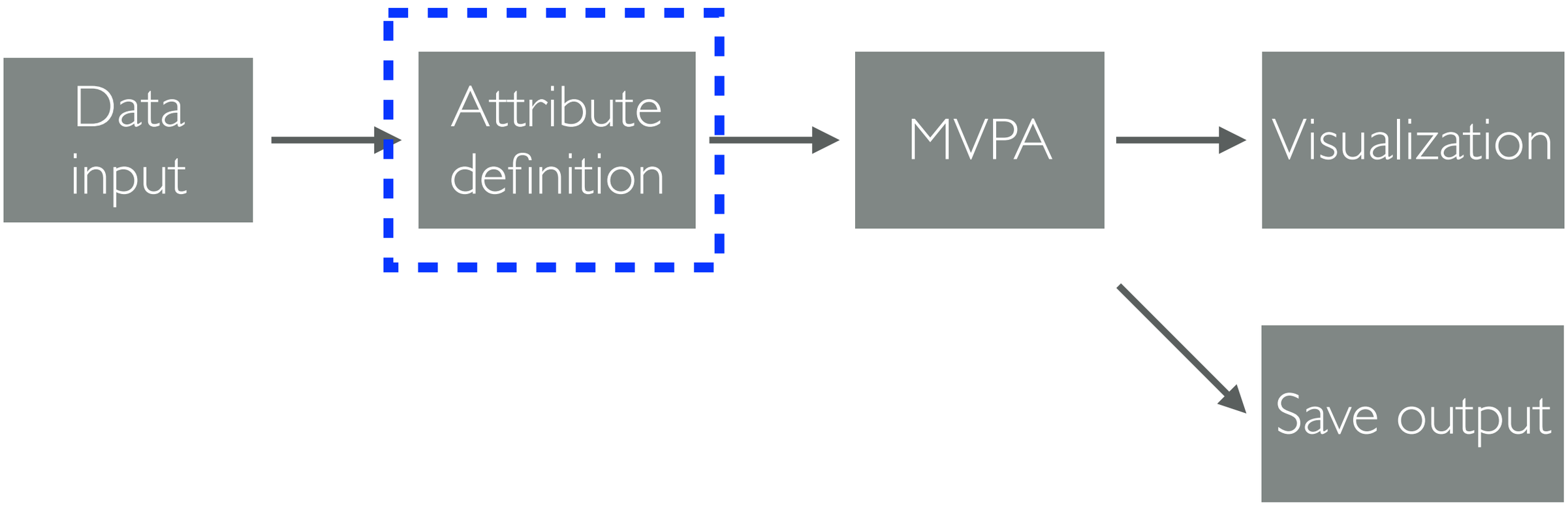
- Data input



http://cosmomvpa.org/index.html

CoSMo MVPA

- Analysis pipeline (within subject)



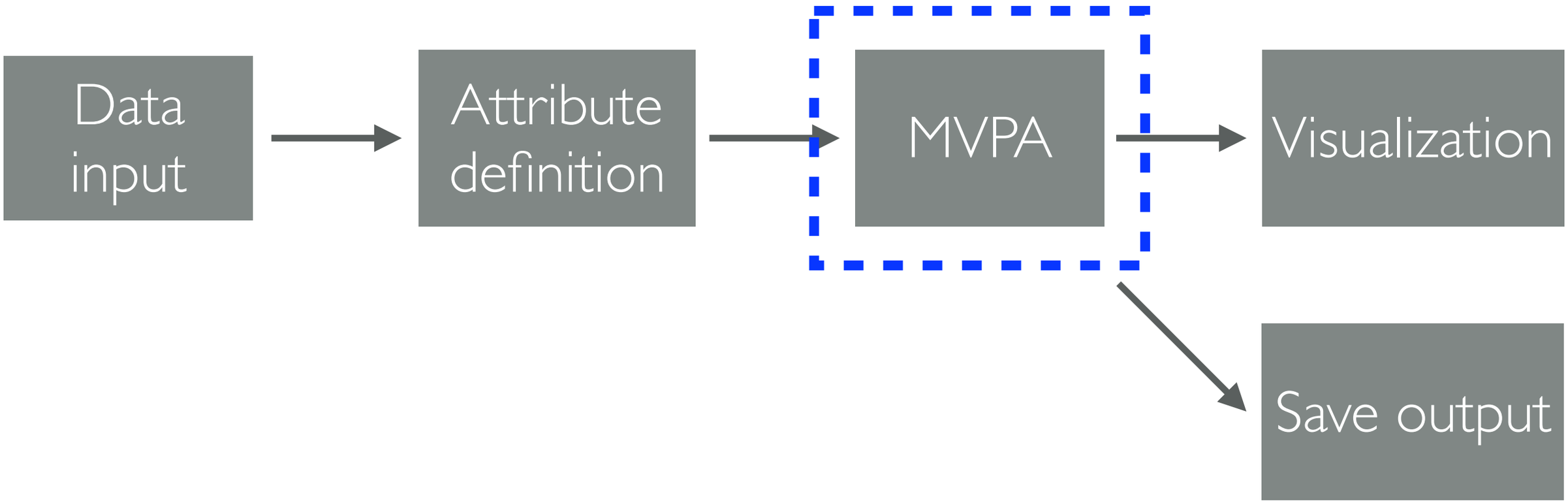
CoSMo MVPA

- Sample attributes:
    - targets
    - chunks
    - labels: actual labels of sample attributes (condition labels as cell array of strings)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Analysis pipeline (within subject)



CoSMo MVPA

- Classifier input:
    - train_samples: training data
    - train_targets: training targets (numeric condition labels)
    - test_samples: testing data
    - opt: additional optional arguments

http://cosmomvpa.org/index.html

CoSMo MVPA

- To split/"partition" data into training/testing
  - Common run partitions for "cross-validation":
    - leave-one-run-out / nfold
    - even/odd

http://cosmomvpa.org/index.html

CoSMo MVPA

- To split/"partition" data into training/testing
  - Can use cosmo_slice (http://cosmomvpa.org/matlab/cosmo_slice.html#cosmo-slice) with cosmo_nfold_partitioner (http://cosmomvpa.org/matlab/cosmo_nfold_partitioner.html) or another scheme (e.g. define odd/even runs)

http://cosmomvpa.org/index.html

CoSMo MVPA

- To split/"partition" data into training/testing
  - Example code:

```
even_msk = mod(ds.sa.chunks,2)==0;
ds_even = cosmo_slice(ds, even_msk);
```

http://cosmomvpa.org/index.html

CoSMo MVPA

- Common classifiers:
    - cosmo_classify_lda
    - cosmo_classify_naive_bayes
    - cosmo_classify_svm

http://cosmomvpa.org/index.html

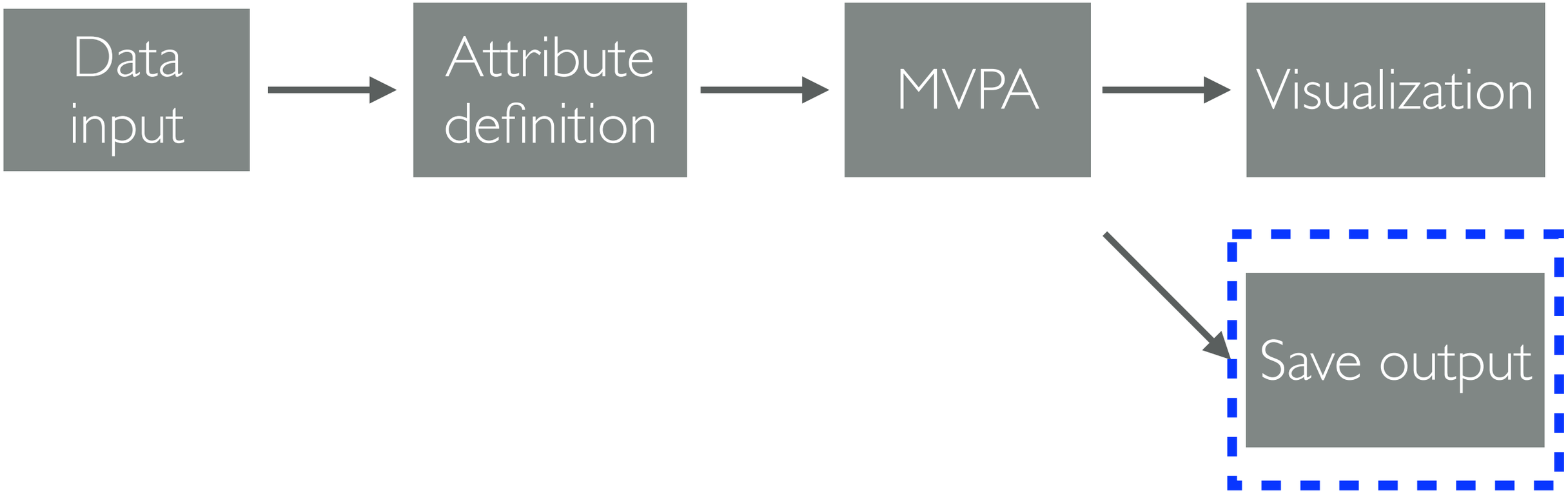
CoSMo MVPA

- Outputs predicted condition labels for each test sample
- Can compare with actual test conditions (targets) to calculate classification accuracy

http://cosmomvpa.org/index.html

CoSMo MVPA

- Analysis pipeline (within subject)



CoSMo MVPA

- Can save results to disk in different file formats
- e.g. save to volumetric Nifti, AFNI, or BV file format depending on ending of file name:
    - cosmo_map2fmri(results, output_fn)

http://cosmomvpa.org/index.html

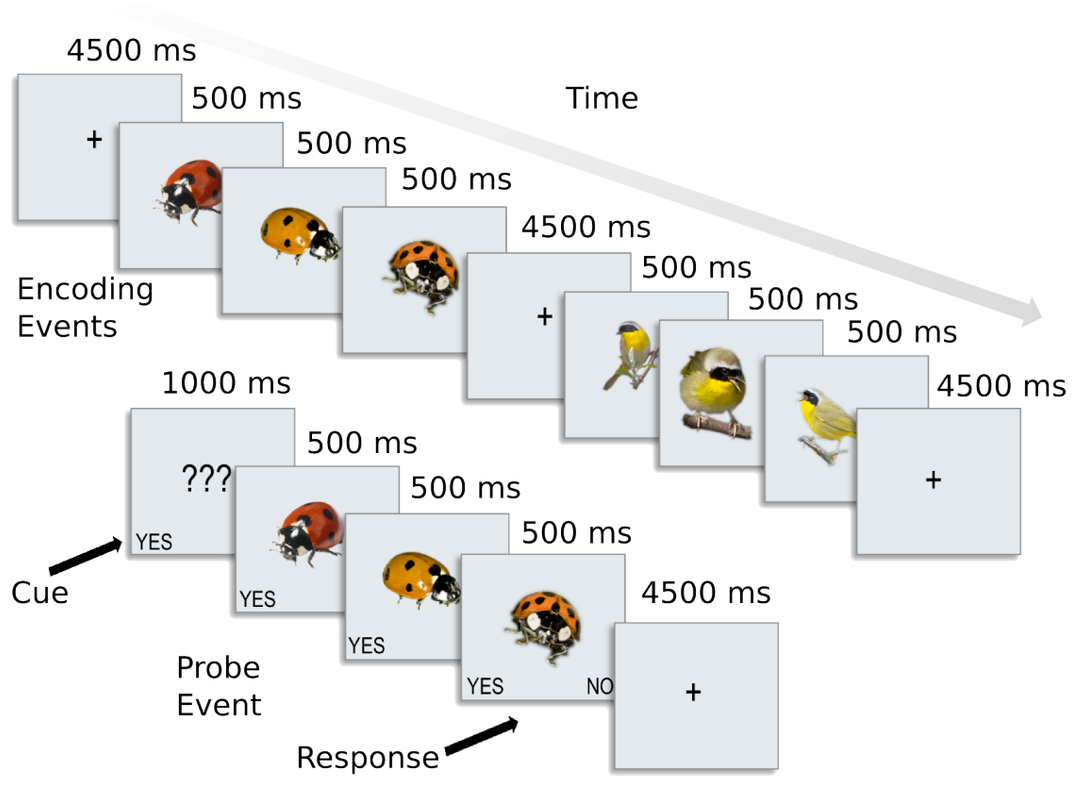
CoSMo MVPA

- ROI-based LDA classification DEMO
  - Based on: http://cosmomvpa.org/matlab/run_classify_lda.html#run-classify-lda
  - Single subject
  - 10 runs
  - 6 conditions: monkey, lemur, mallard, warbler, ladybug, lunamoth

http://cosmomvpa.org/index.html

CoSMo MVPA

- ROI-based LDA classification DEMO



http://cosmomvpa.org/index.html

CoSMo MVPA

- ROI-based LDA classification DEMO
  - GLM to estimate responses for each category in each run
  - Use resulting t-values for each condition/run as input to classification

http://cosmomvpa.org/index.html

CoSMo MVPA

- See demo_classify_lda.m

http://cosmomvpa.org/index.html

CoSMo MVPA

- Searchlight analysis:
  - Mask input is a binary whole-brain mask
  - Size of searchlight

http://cosmomvpa.org/index.html

CoSMo MVPA

- Searchlight analysis:
  - Only requires a few extra steps:
    - neighborhood (voxel coordinates of each searchlight)
    - measure_args (partitions)
    - (measure (classification type) if using the wrapper cosmo_searchlight function)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Searchlight analysis:
  - Define neighborhoods; nbrhood = :
    - cosmo_spherical_neighborhood(ds, 'radius', radius)
      - Searchlight radius in number of voxels
      - *truncates searchlights at edge of brain
    - cosmo_spherical_neighborhood(ds, 'count', vox_count)
      - Searchlight of vox_count number of voxels total

http://cosmomvpa.org/index.html

CoSMo MVPA

- Searchlight analysis:
    - Set partition scheme structure, with cell arrays of indices for each partition
    - e.g. : measure_args.partitions = cosmo_oddeven_partitioner(ds)
        - cosmo_nfold_partitioner
        - cosmo_nchoosek_partitioner

http://cosmomvpa.org/index.html

CoSMo MVPA

- Searchlight analysis:
  - Set measurement type (only for cosmo_searchlight wrapper), e.g. :
    - measure = @cosmo_crossvalidation_measure
      - @cosmo_target_dsm_corr_measure

http://cosmomvpa.org/index.html

CoSMo MVPA

- Searchlight analysis:
  - Set classifier type (only for cosmo_searchlight wrapper), e.g. :
    - measure_args.classifier = @cosmo_classify_lda
      - @cosmo_classify_naive_bayes
      - @cosmo_classify_svm

http://cosmomvpa.org/index.html

CoSMo MVPA

- Searchlight analysis input:
    - dataset structure (ds)
    - neighborhoods (nbrhood)
    - (measure (classification type))
    - partitions (measure_args)

http://cosmomvpa.org/index.html

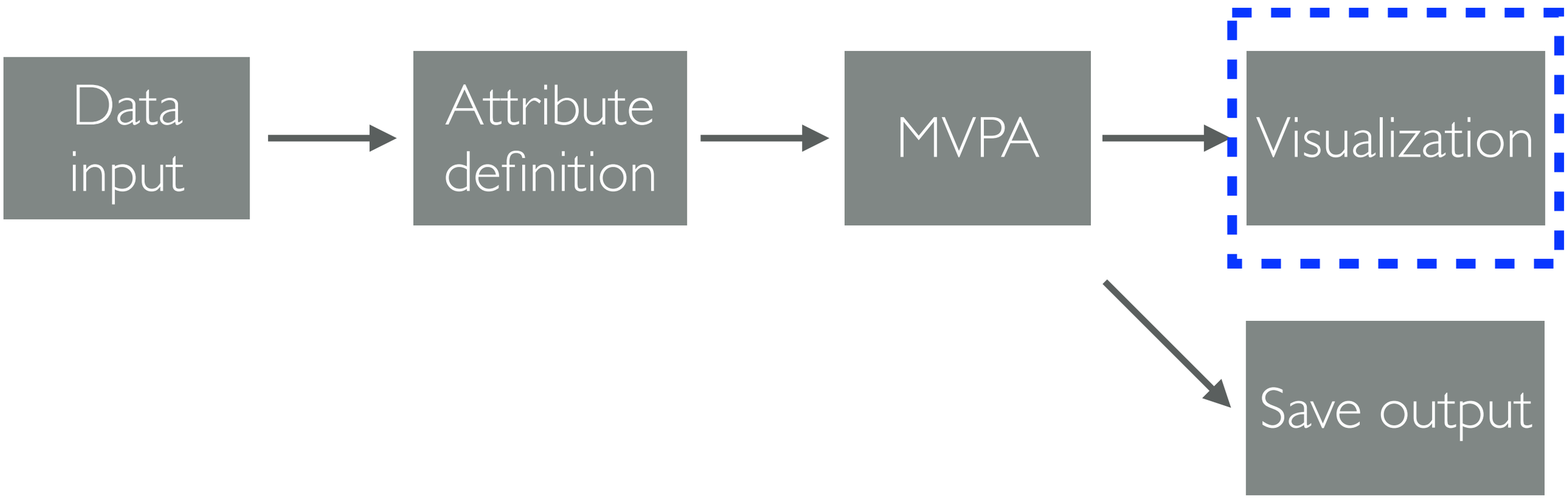
CoSMo MVPA

- Searchlight analysis:
  - Wrapper: cosmo_searchlight(ds, nbrhood, measure, measure_args)
    - cosmo_naive_bayes_classifier_searchlight(ds, nbrhood, measure_args)
  - Outputs a whole-brain map of classification accuracies

http://cosmomvpa.org/index.html

CoSMo MVPA

- Analysis pipeline (within subject)



CoSMo MVPA

- Plot results slice-by-slice:
    - cosmo_plot_slices(results)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Naive bayes searchlight DEMO
  - Single subject
  - 4 runs, 4 blocks per run
  - 2 conditions: index finger press, middle finger press

http://cosmomvpa.org/index.html

CoSMo MVPA

- See demo_searchlight_naive_bayes.m

http://cosmomvpa.org/index.html

CoSMo MVPA

- Representational similarity analysis
    - Load data
    - Compute average pattern for each condition (across runs)
    - Compute similarity for each pair of conditions

http://cosmomvpa.org/index.html

CoSMo MVPA

- Representational similarity analysis
    - Compute average pattern for each condition (across runs)
        - cosmo_fx(ds, @(x)mean(x,1), 'targets', 1)
            - 2nd input is function to apply
            - 3rd input is features to split by
            - 4th input is dimension to split on (1=samples, 2=features)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Representational similarity analysis
    - Compute similarity for each pair of conditions
        - dsm = cosmo_pdist(ds.samples, 'correlation');
            - computes pairwise distance between samples in a matrix

http://cosmomvpa.org/index.html

CoSMo MVPA

- Representational similarity analysis
    - Can compare neural similarity matrix with one from another domain (e.g. behavior) using cross-correlation
    - cc = cosmo_corr(dsms')
    - Computes Pearson correlation between DSMs

http://cosmomvpa.org/index.html

CoSMo MVPA

- Representational similarity analysis DEMO
  - 8 subjects
  - 2 masks (early visual, ventral temporal)
  - 6 conditions: monkey, lemur, mallard, warbler, ladybug, lunamoth

http://cosmomvpa.org/index.html

CoSMo MVPA

- Representational similarity analysis DEMO
  - For each subject & mask, compute neural dissimilarity matrix
  - Then compare these with a V1 model-based dissimilarity matrix, and a behavioral dissimilarity matrix

http://cosmomvpa.org/index.html

CoSMo MVPA

- see demo_similarity_analysis.m

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing
    - Cluster-based statistics
        - one- or two-sample t-test
        - one-way or repeated measures ANOVA
- Can be used for individual subject, or for group-level testing (depending on how targets/chunks options are specified; see function info http://cosmomvpa.org/matlab/cosmo_montecarlo_cluster_stat_hdr.html)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing
    - Used to find significant voxel clusters, while correcting for multiple comparisons
        - Size-based
        - Number-based
        - TFCE

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing
  - cosmo_montecarlo_cluster_stat(ds, nbrhood, […])
  - http://cosmomvpa.org/matlab/cosmo_montecarlo_cluster_stat_hdr.html

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing
  - Default is to use a sign-flipping method of ds data to create null data (conservative)
  - Can also input generated null data (e.g. using cosmo_randomize_targets to shuffle condition labels and run original analysis n times)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing with TFCE for group-level statistics of searchlight classification results
- Register each subject's classification map to standard space
- Merge results into a single file, in which each row corresponds to one subject's data (each column is a voxel)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing with TFCE for group-level statistics of searchlight classification results

- For each subject, re-run the classification analysis 50-100 times, but with randomly shuffled condition labels
  - Take original code, put searchlight into a loop & iterate
- Merge each iteration results into single group file (so have 50-100 group results based on randomized labels)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing with TFCE for group-level statistics of searchlight classification results
  - Load null data into a cell array, such that each cell contains a dataset structure
  - Run cosmo_montecarlo_cluster_stat with null data argument

http://cosmomvpa.org/index.html

CoSMo MVPA

- Monte Carlo permutation testing with TFCE for group-level statistics of searchlight classification results
- Function:
  - Computes map of TFCE values based on original results
  - Computes maps of TFCE values based on null data
  - For each voxel, compares original value to distribution of values based on null data (resulting in a corrected probability value of observation vs. chance)

http://cosmomvpa.org/index.html

CoSMo MVPA

- see group_multiple_comparisons_nulldata_1tail.m
  - (this code is not runnable b/c the datasets are not located in the workshop directory, but is provided for your reference)

- also, group_mean.m for computing mean of searchlight accuracy maps across subjects

http://cosmomvpa.org/index.html

CoSMo MVPA

- see demo_tfce.m
- (this code is runnable)

http://cosmomvpa.org/index.html

CoSMo MVPA

- Today's exercise:
  - RSA searchlight: http://cosmomvpa.org/ex_rsa_tutorial.html
- Other exercises for practice:
  - http://cosmomvpa.org/labman2017_ex_toc.html

http://cosmomvpa.org/index.html

CoSMo MVPA

- List of functions:
  - http://cosmomvpa.org/matindex.html
- More demos:
  - http://cosmomvpa.org/contents_demo.html#demo-fmri-searchlight-lda
- More runnable examples:
  - http://www.cosmomvpa.org/contents_demo.html#id1

http://cosmomvpa.org/index.html

CoSMo MVPA

- Be sure to cite!

  - Oosterhof, N. N., Connolly, A. C., and Haxby, J.V. (2016). CoSMoMVPA: multi-modal multivariate pattern analysis of neuroimaging data in Matlab / GNU Octave. Frontiers in Neuroinformatics, doi:10.3389/fninf.2016.00027.

http://cosmomvpa.org/index.html