http://neuro.debian.net

neurodebian

PyMVPA

http://www.pymvpa.org

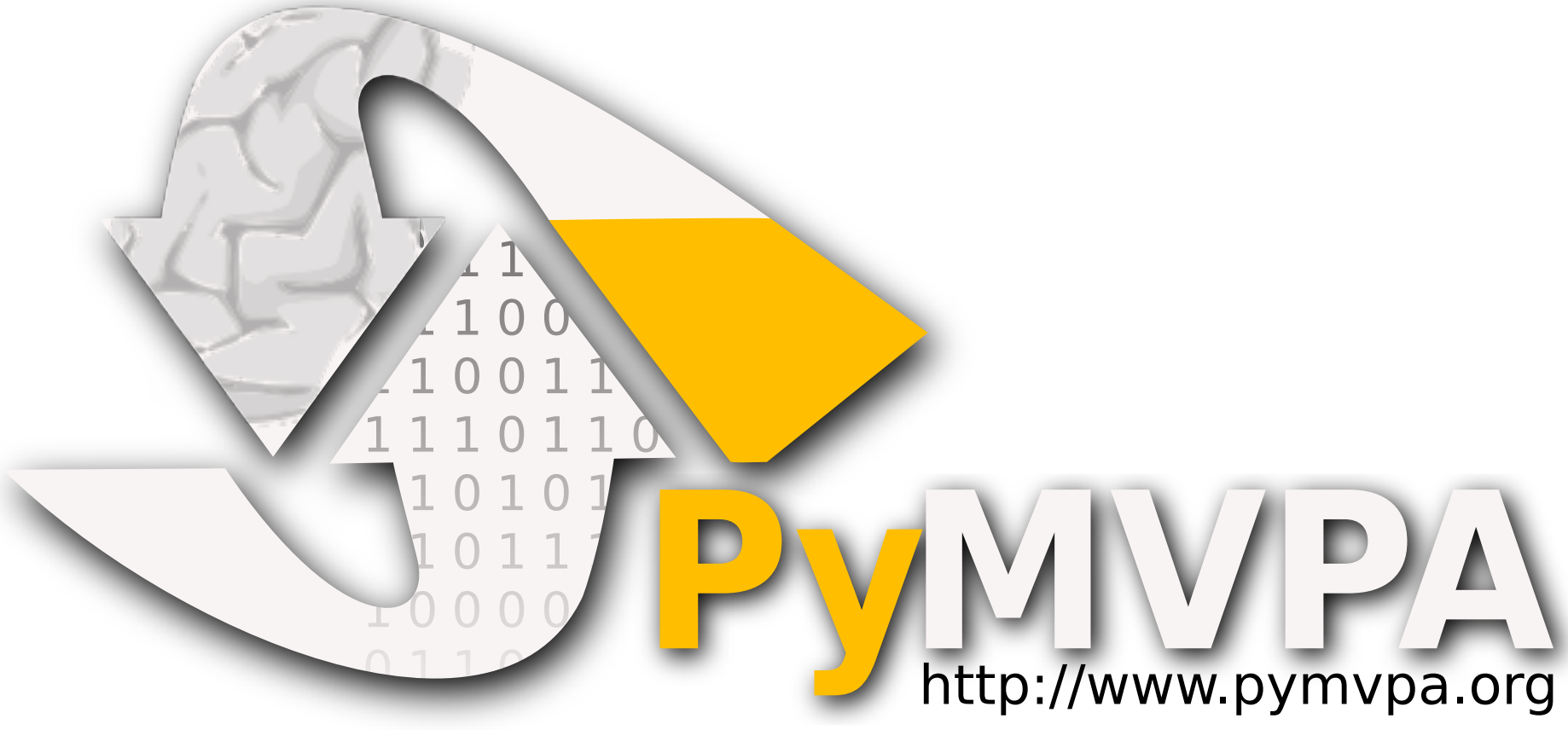
Overview of statistical evaluation techniques adopted by publicly available MVPA toolboxes

Yaroslav O. Halchenko

Center for Cognitive Neuroscience, Psychological and Brain Sciences Department,
Dartmouth College
PyMVPA, NeuroDebian, DataLad, duecredit

OHBM 2015, Honolulu HI

Disclaimer #1: I do PyMVPA



Many thanks to

- Andre Marquand (PROBID, Matlab)
- Francisco Pereira (searchmight, Matlab)
- Gael Varoquaux (nilearn/scikit-learn, Python)
- Jessica Schrouff (PRoNTo, Matlab)
- Martin Hebart (TDT, Matlab)
- Mitsuaki Tsukamoto (BDTB, Matlab)
- Nick Oosterhof (CoSMoMVPA, Matlab)
- Nikolaus Kriegeskorte (RSA, Matlab) [will not talk about]

Many thanks to

- Andre Marquand (PROBID, Matlab)
- Francisco Pereira (searchmight, Matlab)
- Gael Varoquaux (nilearn/scikit-learn, Python)
- Jessica Schrouff (PRoNTo, Matlab)
- Martin Hebart (TDT, Matlab)
- Mitsuaki Tsukamoto (BDTB, Matlab)
- Nick Oosterhof (CoSMoMVPA, Matlab)
- Nikolaus Kriegeskorte (RSA, Matlab) [will not talk about]

Q: who uses any of the aforementioned toolkits (including PyMVPA)?

Many thanks to

- Andre Marquand (PROBID, Matlab)
- Francisco Pereira (searchmight, Matlab)
- Gael Varoquaux (nilearn/scikit-learn, Python)
- Jessica Schrouff (PRoNTo, Matlab)
- Martin Hebart (TDT, Matlab)
- Mitsuaki Tsukamoto (BDTB, Matlab)
- Nick Oosterhof (CoSMoMVPA, Matlab)
- Nikolaus Kriegeskorte (RSA, Matlab) [will not talk about]

Q: who uses any of the aforementioned toolkits (including PyMVPA)?
Q: who uses some other (not your own) toolkit?

Many thanks to

- Andre Marquand (PROBID, Matlab)
- Francisco Pereira (searchmight, Matlab)
- Gael Varoquaux (nilearn/scikit-learn, Python)
- Jessica Schrouff (PRoNTo, Matlab)
- Martin Hebart (TDT, Matlab)
- Mitsuaki Tsukamoto (BDTB, Matlab)
- Nick Oosterhof (CoSMoMVPA, Matlab)
- Nikolaus Kriegeskorte (RSA, Matlab) [will not talk about]

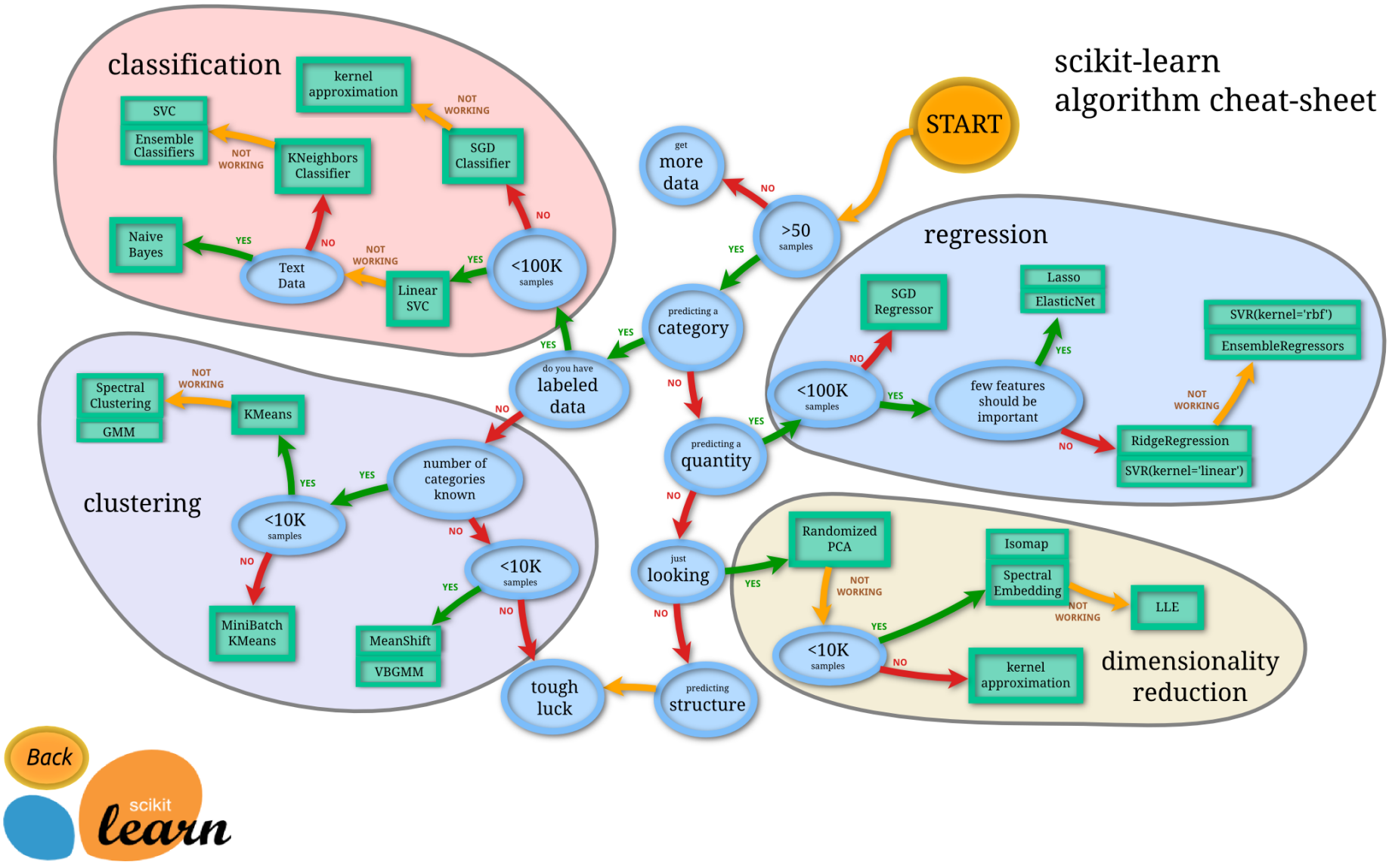
Q: who uses any of the aforementioned toolkits (including PyMVPA)?
Q: who uses some other (not your own) toolkit?
Q: who writes ”ad-hoc” code instead?

Disclaimer #2:

This review is not comprehensive

- I will note where and what functionality *is* available.
  I will *not* state that some functionality is not available in a particular toolbox
- I will *not* provide overview of interfaces (e.g. scripting vs. GUI), but rather will reference the functionality available
- I will *not* talk about ”sensitivities” analyses although many toolboxes allow for some

We (as in neuroimaging) are special



We (as in neuroimaging) are special

Machine-learning folks

- construct the **best** predictive model given a large array of samples
- characterize the model by accuracy of classification on some **canonical** datasets

We (as in neuroimaging) are special

Machine-learning folks

- construct the **best** predictive model given a large array of samples
- characterize the model by accuracy of classification on some **canonical** datasets

We (Neuroimaging) folks

- construct a model **good enough** to state that data contain information of interest
- use summary statistic computed over obtained accuracies to support claim of presence of the signal of interest in **new** dataset

Additional “support measures”

- Gut feeling (https://en.wikipedia.org/wiki/Feeling)

Additional “support measures”

- Gut feeling (https://en.wikipedia.org/wiki/Feeling)
  - Priors (expertise, publications, NeuroSynth.org)

Additional “support measures”

- Gut feeling (https://en.wikipedia.org/wiki/Feeling)
  - Priors (expertise, publications, NeuroSynth.org)
- Are results *trustworthy*?

Additional “support measures”

- Gut feeling (https://en.wikipedia.org/wiki/Feeling)
  - Priors (expertise, publications, NeuroSynth.org)
- Are results *trustworthy*?
  - Stable
  - Reproducible
  - Not “random”

Beliefs

Fisher's Beliefs Regarding p Values	
p value	Fisher's statements
.1 to .9	"Certainly no reason to suspect the hypothesis tested" (p. 79)
.02 to .05	"Judged significant, though barely so . . . these data do not, however, demonstrate the point beyond possibility of doubt" (p. 122)
below .02	"Strongly indicated that the hypothesis fails to account for the whole of the facts" (p. 79)
below .01	"No practical importance whether p is .01 or .000001" (p. 89)

see "Dance of the p Values"

https://www.youtube.com/watch?v=5OL1RqHrZQ8

Wright, D. B. (2009). Ten statisticians and their impacts for psychologists. *Perspectives on Psychological Science*, 4(6):587–597

Statistical significance testing can improve the ”level of trust” in observed results

Factors . . . affecting ”level of trust”

negatively

- Software bugs [Do you trust your tools?]
- Experimental design *bugs*
- Analysis *bugs* I: double dipping
- Analysis *bugs* II: exploitation

Factors . . . affecting ”level of trust”

negatively

- Software bugs [Do you trust your tools?]
- Experimental design *bugs*
- Analysis *bugs* I: double dipping
- Analysis *bugs* II: exploitation

positively

- Statistical significance of the results

Factors . . . affecting ”level of trust”

negatively

- Software bugs [Do you trust your tools?]
- Experimental design *bugs*
- Analysis *bugs* I: double dipping
- Analysis *bugs* II: exploitation

positively

- Statistical significance of the results

All of the above is not MVPA-specific, but

”With great power comes great responsibility” (Uncle Ben)

Under assumption of bug-free implementation, how can existing toolboxes help to improve the “level of trust” in our MVPA results?

Experimental design *bugs*

Major manifestations

- Imbalances
- Trial order effects

Experimental design *bugs*

Scrutinize design (per subject)

- Review labeling stats: PyMVPA (`dataset.summary()`, includes trial order stats), PRoNTo
- “Decode” the design: TDT (not based on trial-order)
- Remove overlaps: PRoNTo

Experimental design *bugs*

Scrutinize design (per subject)

- Review labeling stats: PyMVPA (`dataset.summary()`, includes trial order stats), PRoNTo
- "Decode" the design: TDT (not based on trial-order)
- Remove overlaps: PRoNTo

Avoid inbalance

- Mean the trials. **Don't!**: introduces spurious signal
- Sub-sample: PyMVPA, CoSMoMVPA (disallows imbalance, and allows re-balancing), TDT, BDTB
- Metrics other than overall accuracy
  - AUC: scikit-learn, PyMVPA, TDT
  - balanced accuracy/mean of per-class accuracies: PRoNTo, PROBID, TDT

Analysis *bugs* I: double dipping (DD), circular analysis

Cross-validation constructs

- Split according to natural confounds (e.g. runs/sessions): all
- Flexible (PyMVPA, nilearn, CoSMoMVPA, TDT) or more restricted (PROBID, searchmight, BDTB) forbidding double-dipping
- Combined with pre-processing, such as feature selection or tranformation (e.g. PCA): PyMVPA, scikit-learn, CoSMoMVPA, TDT, PRoNTo

Analysis *bugs* I: double dipping (DD), circular analysis

Cross-validation constructs

- Split according to natural confounds (e.g. runs/sessions): all
- Flexible (PyMVPA, nilearn, CoSMoMVPA, TDT) or more restricted (PROBID, searchmight, BDTB) forbidding double-dipping
- Combined with pre-processing, such as feature selection or tranformation (e.g. PCA): PyMVPA, scikit-learn, CoSMoMVPA, TDT, PRoNTo

Nested cross-validation

- Parameter selection: scikit-learn, PRoNTo, TDT, (PyMVPA on example, convenience – coming)
- Recursive feature selection/elimination: PyMVPA, PROBID, scikit-learn, TDT

Analysis *bugs* II: exploitation of the models

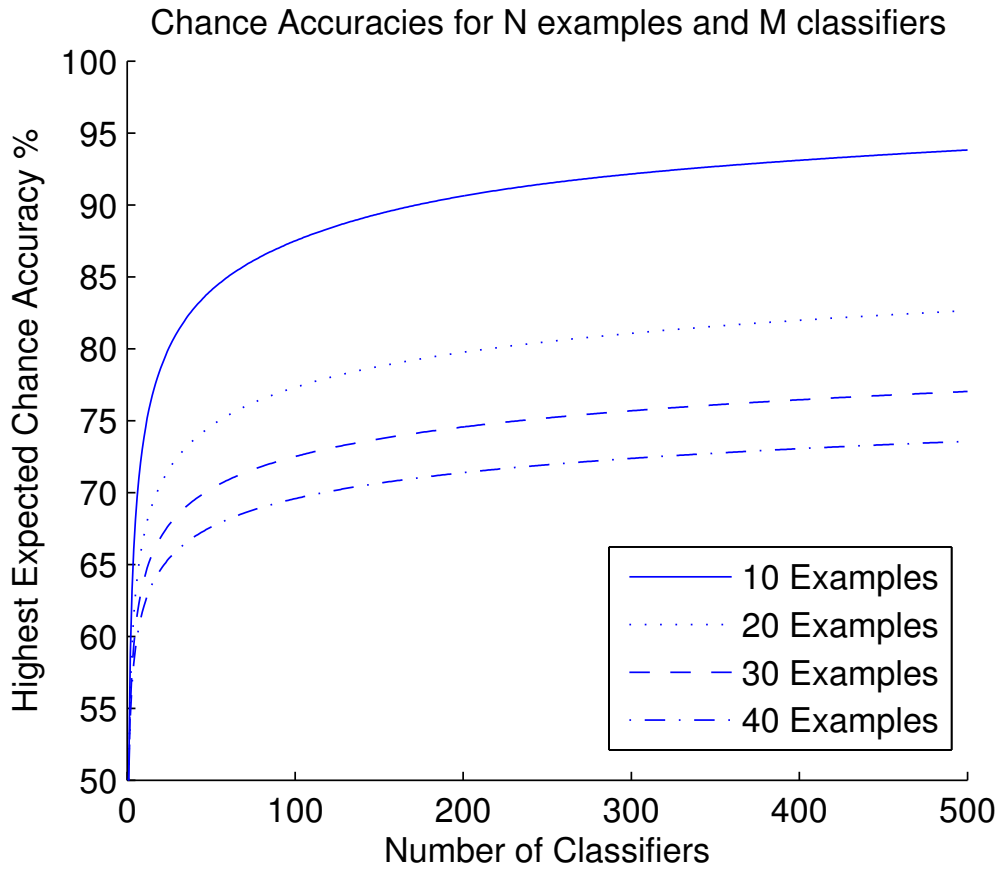


Figure 1. The highest expected chance accuracy as a function of the number of examples and classifiers. Each line represents a different number of examples. The x-axis is the number of classifiers and the y-axis is the accuracy.

Palatucci, M. and Carlson, A. (2008). On the chance accuracies of large collections of classifiers. In *Proceedings of the 25th International Conference on Machine Learning*

Analysis *bugs* II: exploitation of the models

Prevention mechanisms

- Nested CV model selection (scikit-learn, PRoNTo, TDT, PyMVPA)
- Some toolboxes restrict variety of available classifiers to mitigate
- Some expose as many as possible to demonstrate it:
  PyMVPA: **clfswh** comes with $>=$ 36 of ready-to-be-abused clfs
  (including a few ”Random” ones)

Analysis *bugs* II: exploitation of the models

Prevention mechanisms

- Nested CV model selection (scikit-learn, PRoNTo, TDT, PyMVPA)
- Some toolboxes restrict variety of available classifiers to mitigate
- Some expose as many as possible to demonstrate it:
  PyMVPA: **clfswh** comes with $>=$ 36 of ready-to-be-abused clfs
  (including a few ”Random” ones)

Recommendations

- Establish the ”best” pipeline on an independent sample/study
- Verify absent “results” on random/unrelated data

Significance estimation

H0 distribution estimation (randomization approaches)

- Dummy classifiers (PyMVPA, nilearn/scikit-learn, PRoNTo)
- Random, from another experiment(s), outside of the brain data
- MC permutation (PyMVPA, nilearn/scikit-learn, CoSMoMVPA, PROBID (2 class), TDT, PRoNTo)
  - a must #1: within each run (we seems to be in clear)
  - a must #2: permutation for all CV folds at once
  - maintaining temporal structure. PyMVPA:
    - maintaining target labeling in test portion only
    - labels reassignment (`strategy='uattrs'`)
    - reassignment of labeling across sessions/chunks (`strategy='chunks'`)
- All can do semi-parametric; PyMVPA can also perform semi-parametric estimation

Significance estimation: searchlights

Make it feasible + multiple comparison problems

- Simple classifiers == fast: GNB/M1NN searchmight, PyMVPA
- Spatial sub-sampling (Björnsdotter et al., 2011): PyMVPA
- Per-subject randomization + bootstrap (Stelzer et al., 2013):
  - PyMVPA (cluster-level with some minor mods + FDR correction on cluster level p's)
  - CoSMoMVPA (cluster-level based, with TFCE correction)
- "Flipping" around chance-level of actual performance metrics to simulate chance distribution of the mean (CoSMoMVPA)

Summary

”With great power comes great responsibility”

—Uncle Ben

- Significance testing should provide ”support” but not the ultimate verdict
- MVPA is/can be more sensitive to experimental design flaws
- Avoid common pitfalls: good randomization of trial orders and scrutiny of the design and results is a must
- Existing MVPA toolboxes provide a complementary spectrum of methodologies to help avoiding pitfalls and provide statistical assessments of the results

Thank you!

References

Björnsdotter, M., Rylander, K., and Wessberg, J. (2011). A monte carlo method for locally multivariate brain mapping. *NeuroImage*, 56(2):508–516.

Palatucci, M. and Carlson, A. (2008). On the chance accuracies of large collections of classifiers. In *Proceedings of the 25th International Conference on Machine Learning*.

Stelzer, J., Chen, Y., and Turner, R. (2013). Statistical inference and multiple testing correction in classification-based multi-voxel pattern analysis (MVPA): Random permutations and cluster size control. *NeuroImage*, 65(0):69 – 82.

Wright, D. B. (2009). Ten statisticians and their impacts for psychologists. *Perspectives on Psychological Science*, 4(6):587–597.