Predictive models in neuroscience and bioinformatics

By

Yuval Benjamini

A dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Bin Yu, Chair
Professor Terence P. Speed
Professor Jack L. Gallant

Spring 2013
Predictive models in neuroscience and bioinformatics

Copyright 2013

by

Yuval Benjamini
Abstract

Predictive models in neuroscience and bioinformatics

by

Yuval Benjamini

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Bin Yu, Chair

This dissertation discusses how predictive models are being used for scientific inquiry. Statistical and computational advances have given rise to high-dimensional models that can be fit on relatively small samples but still predict well the behavior of complex systems. Scientists try to use such models to learn about complex biological systems; but it is not always clear how prediction accuracy translates to understanding the underlying system. In the chapters below, I present different approaches to learn from predictive models in bioinformatics and neuroscience. In each of these collaborative works, we tailor models that would both fit well and be interpretable in the context of the scientific questions.

In the first chapter, we fit and compare predictive models for the GC-content bias, an important confounder in DNA-sequencing. We develop a high-resolution model that treats each base-pair in the genome as a separate example; this allows us to compare many representations of GC-content, identifying which representation best predicts the variation in the coverage. To deal with the huge volumes of data, we develop a new conditional dependence measure that efficiently compares different models. Selection of the model that maximizes this dependence reveals a recurring association with an experimental parameter: the selected model in each sample corresponds to a window size almost identical to the average size of DNA fragments in the sample. This recurring result can be used both for correcting the bias and for learning about the causes for the bias.

In the next chapter, we propose a new estimator for interpreting prediction-accuracy results of models for neural activity in the visual cortex. Our shuffle estimator targets the explainable
variance - the proportion of signal in the measured response - while accounting for autocorrelation in the noise. Re-analyzing models of functional MRI voxels within visual area V1, we observe a strong linear correlation between the signal-to-noise and prediction accuracy.

In the final chapter we analyze neurophysiology data recorded from visual area V4, and present a full cycle of scientific investigation using prediction models in neuroscience. Whereas the previous chapters developed metrics for evaluating feature sets and prediction models, this chapter takes an extra leap: we use optimization algorithms together with prior scientific knowledge to propose a new feature-set. We then fit regularized linear models based on this representation that generalize well to a validation data set. Finally, novel visualization and model-summary techniques help interpret the resulting prediction models, revealing rich patterns of activity in the different neurons and unexpected categories of neurons.
The best thing about being a statistician is that you get to play in everyone’s backyard. (John Tukey).
Contents

List of Figures vi
List of Tables vii
Acknowledgments viii

1 Introduction 1

2 Summarizing and correcting the GC-content bias in high throughput sequencing 6
   2.1 Introduction ........................................ 7
   2.2 Materials and Methods ................................ 9
       2.2.1 Mapped read files ................................ 9
       2.2.2 Loess model ..................................... 10
       2.2.3 Single position models ................................ 11
       2.2.4 Evaluation ........................................ 14
       2.2.5 Software and data availability ...................... 15
   2.3 Results .............................................. 16
       2.3.1 Bin Counts ........................................ 16
       2.3.2 Single position models ............................. 17
       2.3.3 Effect of fragment length ........................ 20
       2.3.4 Local biases near fragment ends .................... 21
       2.3.5 Aggregating effects and corrections ................. 21
       2.3.6 Poisson and other variation ......................... 24
       2.3.7 Additional data sets ............................... 26
   2.4 Discussion ............................................. 27
   2.5 Supplementary Data .................................... 30

3 The shuffle estimator for explainable variance in fMRI experiments 31
   3.1 Introduction .......................................... 32
   3.2 Preliminaries .......................................... 35
       3.2.1 An FMRI Experiment ............................... 35
3.2.2 Correlation in the data ........................................ 37
3.2.3 A probability model for the measurements. ................. 39
3.2.4 Explainable variance and variance components ............... 41
3.2.5 Quadratic contrasts .......................................... 43
3.3 Shuffle estimators for signal and noise variances .......... 44
3.3.1 Noise conserving permutation for Y with respect to design X 45
3.3.2 Shuffle estimators ........................................ 47
3.4 Evaluating prediction for correlated responses ............... 50
3.5 Simulation .................................................... 51
3.5.1 Block structure ........................................... 51
3.5.2 Time-series model ......................................... 52
3.5.3 Comparison to REML ..................................... 53
3.5.4 Results .................................................... 54
3.6 Data .......................................................... 54
3.6.1 Methods ................................................... 54
3.6.2 Results .................................................... 55
3.7 Discussion ...................................................... 57
3.8 Appendix ........................................................ 59
3.8.1 Consistency of $\omega^2$ .................................. 59
3.8.2 Lemma (2) ................................................ 63

4 Modeling V4 under Naturalistic Conditions with Invariant Image Representations 66
4.1 Introduction ..................................................... 67
4.2 Results ........................................................ 68
4.2.1 Hierarchical Image Feature Extraction for V4 Based on Invariance and Sparse Coding Principles ......................... 69
4.2.2 Dictionary Learning on Orientation Map Patches from Natural Images Produces Complex Interpretable Features ................. 71
4.2.3 Prediction Performance in V4 Is as Good as in Earlier Areas V1 and V2. ............................................. 73
4.2.4 Individual Neurons in V4 Respond to Diverse Stimuli Characteristics ................................................... 76
4.2.5 V4 Neurons Cluster Into Two Groups: Those Selective to Image Contours Versus Those Selective to Texture .................. 79
4.3 Discussion ...................................................... 82
4.3.1 Relation With Previous Neurophysiological Studies Using Artificial Stimuli .................................................. 82
4.3.2 Relations With Previous Neuronal Models .................. 82
4.3.3 Are the Receptive Fields Space-Time Separable? ........... 83
4.3.4 Other Roles of V4: Color and Motion ....................... 83
4.3.5 Biological Compatibility and Links With Computer Vision Models ................................................... 84
4.4 Methods ........................................................ 84
4.4.1 Experimental Procedures ................................... 84
4.4.2 An Invariant Computational Model for V4 Tuned to Natural Images . 86
4.4.3 Prediction model - learning semi time-separable models with low-rank
regularization. .......................................................... 88
4.4.4 Inferring Neuron Selectivity From Fitted Prediction Model . . . . . 89
4.4.5 Estimation of explainable variance ........................................ 92

Bibliography ................................................................. 94
## List of Figures

2.1 Single position model estimation ......................................................... 11
2.2 GC curves (10 kb bins) ........................................................................ 16
2.3 Single position models .......................................................................... 18
2.4 Different lags ........................................................................................ 19
2.5 Fragment rate by length and GC ............................................................ 20
2.6 Fragmentation effect .............................................................................. 22
2.7 Aggregation of single Location estimates ............................................... 23
2.8 Corrected counts of the normal sample .................................................. 24
2.9 Copy number gains of the tumor sample ................................................. 25
2.10 Comparison to Poisson variation ............................................................. 26
2.11 GC plots for Dataset 2 .......................................................................... 27

3.1 A Gabor-based encoding model for natural images ................................. 37
3.2 Prediction accuracy for V1 voxels ............................................................. 38
3.3 Data acquisition for the validation data set .............................................. 40
3.4 Cartoon of the shuffle estimator .............................................................. 45
3.5 Simulations for the block and time-series ............................................... 52
3.6 Comparison of methods on simulation ................................................. 53
3.7 Optimal vs. observed prediction accuracy .............................................. 56
3.8 Comparison of voxels from two functional visual areas: V1 and V4 ........ 57

4.1 Hierarchical image processing in our computational model .................. 70
4.2 Visualization of Complex Features ......................................................... 72
4.3 Predictive Model of Neuron Responses via Low-Rank Regularization .... 74
4.4 Predictive Performance of Our Model on a Test Set ............................... 75
4.5 Summary of the model for one neuron .................................................... 77
4.6 Summary of three neuron models .......................................................... 79
4.7 Texture versus non-texture clusters ....................................................... 80
4.8 Leading components of variation between neuron models .................... 81
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>List of data sets analyzed in this chapter</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Prediction error (MAD) of loess model for different resolutions</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>Residual variance from different models</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Classification of feature (counts)</td>
<td>76</td>
</tr>
</tbody>
</table>
First and foremost, I would like to express my deep gratitude to my advisors, Professor Bin Yu and Professor Terry Speed. Both taught the applied statistic sequence in my first year at Berkeley; by the end of that year, I have found in statistics what I was looking for. To Bin I am thankful for caring and guiding me on a day-to-day basis through the mazes of the academic world, and for helping me address my strengths and weaknesses. To Terry I am thankful for being so generous with time and guidance, and for encouraging me to be myself. Finally, I’d like to thank both for keeping up with what must have been a sub-optimal arrangement for them; I feel extremely privileged to have had both as mentors.

I owe special thanks to Prof. Jack Gallant and the many members of his lab, who invited me to scour their data and their group meetings too. My neuroscience work was built on the solid foundations they have developed over many years. In particular, Dr. Shinji Nishimoto, Dr. Thomas Nasalaris, and Dr. An Vu have provided me priceless guidance. I would like to further mention Prof. Philip Stark from the Statistics Department who provided a friendly ear whenever I needed one.

The chapter discussing the V4 visual area owes much to the work of Julien Mairal; Julien’s contributions were cardinal to the project’s success, and without them there wouldn’t be much to talk about. I am grateful to Julien and to the other coauthors for allowing me to reproduce our work as part of this dissertation. I hope our joint work would continue in the future. The Shuffle estimator chapter benefitted from comments of editors and reviewers in the Annals of Applied Statistics. Oleg Mayba, Su Yeon Kim, and Pierre Neuvial helped me throughout the research underlying the GC-content chapter, and Mark Robinson and Paul Spellman provided much needed expertise. Editing and comments by the editors and reviewers for the Nucleic Acids Research are appreciated. My Berkeley studies where partially funded by the US government through the NSF VIGRE program, the NSF Center for Science of Information, and The Cancer Genome Atlas (TCGA); the american public’s support of the sciences is incredibly generous and has made an impact on my research output.

Academic life of a PhD is intrinsically lonely, but my peers have helped keep my spirits high. I am extremely thankful to my classmates and office mates being such a friendly company. I
know I will miss Sean Ruddy and Sharmodeep Bhattacharyya, my office mates. The frisbee and soccer teams of SGSA provided a welcome excuse to socialize. Furthermore, having two advisors also meant being a part of two research groups, perhaps the two most active in the department. I’d like to thank all members of the Yu group, both young and old, for our work and discussions; and all members of the Speed, Purdom and Dudoit group for providing a European colony inside Berkeley. A special thanks is also in place to the dedicated crew at the Berkeley Espresso, who have made it possible for me to start each morning on the right foot.

This next to last line is dedicated to my father, who stayed with us for half a year in Berkeley. Trying to walk in the footsteps of one’s father has been the natural course in many societies for thousands of years; in today’s world, though, it is not as common. My work ended up following a path extremely close to the one he is now paving. The math is different, but the landscape is the same. I could not have wished a better personal guide than him. My mother served as the main oxygen and information line to the life we have in Israel; Toda Aba veIma.

Finally, none of my achievements would have been made without the help of my wife Ayelet. In addition to the thesis, this refers to Noga and Daphnah, both Berkeley born. Any typos can be blamed on Daphnah, who was born four days before this dissertation was submitted. At this point, I’m sure she would mind.
Chapter 1

Introduction

As I write this dissertation, automated prediction models have taken a prominent role in our everyday lives: they identify our friends, recommend movies for us to see, and even outperform the best human competitors in predicting the answer (rather the question) for a Jeopardy cue. Such models are transforming science as well. Automated off-the-shelf learning algorithms can evaluate the joint effects of more predictors, over many more samples, when compared with a human scientist. These properties make them attractive candidates for modeling the complex and stochastic systems that underlie many biology processes. Furthermore, computational (“dry-lab”) scientists that re-analyze data generated by others rely on predictive models: the validity of an analysis increases if the underlying model can predict well. All this raises the question: if off-the-shelf algorithms can automatically perform the analysis - fit accurate predictive models - what is then the role of the statistician?

Before proposing answers, let us consider the technological shift towards big data in the biological sciences. High-throughput and imaging technologies allow scientists to measure numerous endpoints simultaneously. For example, high-throughput sequencing assays estimate the expression of all human genes in a sample in a single run, by automatically sequencing millions of RNA fragments and matching these fragments to known transcripts. In neuroscience, functional MRI (fMRI) scanners measure online brain-activity along a spatial grid that spans the full brain, and electrode arrays collect electrophysiology information from hundreds of neurons jointly. Advances in digital image processing, the falling price of data storage, and increase in the processing power of personal computer allow individual researchers to access and reanalyze such data from its raw format.

These technological changes initiated many large data-gathering experiments with the goal of mapping and forming hypotheses rather than only confirming existing ones. The data I discuss in this thesis come from such large-scale data collection ventures:
• Most of the high-throughput sequencing data were produced under The Cancer Genome Atlas (TCGA) consortium. The mission of TCGA is to profile tumor cells, grouped by tumor type, with the hope of characterizing indicators and potential causes for cancer.

• The vision data were collected by Jack Gallant’s vision lab in UC Berkeley. The Gallant lab records neural activities in the visual cortex of humans or primates as the subjects perform observation tasks. Their goal is understand the process by which visual information is processed in each of a hierarchy of visual processing areas in the cortex.

In both cases, the analyses are not limited to measuring the effects of a pre-specified set of parameters on several well-defined responses. Rather, the data is collected so to allow analysts as much flexibility as possible to later define what representations of, say, images best reflect the observed neural responses.

Predictive models offer a criterion to overcome this ambiguity in the modeling: a good representation is one that can accurately and consistently predict. In particular, when several representations are available, the prediction criterion is an objective metric, that easily compares across different models approaches.

However, flexibility is not only a blessing, and prediction is not always the best indication of a model’s accuracy. In underdetermined, systems there are often alternative ways of forming models that would predict equally well. We further know that, depending on the noise level and model complexity, the model generating the data can sometimes provide worse predictions than smaller or larger models. Where prediction accuracy is the only goal, as often is the case in engineering applications, optimizing for prediction does not hurt. In science, however, the true goal is often extracting knowledge from the data. Wielding powerful prediction algorithms, it is tempting to show predictive results instead of understanding. How should predictive models be yielded to reach understanding? This is a question we repeatedly tackle in the chapters below.

My short answer is this: predictive models, in science, should be viewed as experiments within a computational framework. As statisticians, we should help design these experiments so that the answers would be most informative in light of the scientific question or claim. The features, the model, the fitting algorithm, and the accuracy metric should all be chosen to help address the question. The statisticians should have both a deep technical understanding of how prediction models work, just as physicists know their lasers: how should models be tailored to optimize prediction in a specific setting, what technical biases should we expect when comparing different models, and what impact would the tuning parameters have. Moreover, they should properly design the prediction experiments, making sure the population of interest can be properly represented by the sample used for the experiment, and that the prediction-accuracy measurement is reliable and valid. Finally, they should quantify the role of chance and structural variability in the results need to assure proper
interpretations.

In the following chapters, I investigate these ideas in the context of three modern biological data-sets. Predictive models are used for reaching scientific conclusions in all three chapters, but serve very different roles. In each, we tailor the feature set, the model, the fitting algorithm and the accuracy metric to the specific properties of the scientific question addressed and the distributional properties of the data.

Here is a brief description of the chapters:

**Chapter 2: DNA sequencing**

High-throughput sequencing is the technology of choice in quantifying the abundance of different genomic regions (DNA) or transcripts (RNA) in a sample. Sequencing is used in numerous experimental protocols in bioinformatics for measuring, for example, gene-expression profiles, affinity to various proteins, and the spatial structure of the chromosomes. This technology employs coverage - the number of mapped fragments covering each base of a reference genome - as a proxy for genomic content: assuming all fragments have equal chance of getting sequenced and mapped, the number of fragments covering each base should reflect the proportions of DNA in the sample. This assumption, however, is too optimistic.

In Chapter 2, originally published as Benjamini and Speed (2012), we refute the assumption of a-priori uniform fragment distribution in many sequencing samples, showing that the distribution is not even consistent across samples. This non-uniformity, termed coverage bias, can confound the signal of interest. We further show that the coverage bias in a genomic region is non-linearly associated with the sequence of bases (A, C, G, and T) that compose the DNA molecule in that region. Models that predict and remove these biases are crucial in order to separate the signal of interest from the sequence-related effects, and to identify the potential sources for the bias.

To evaluate what aspects of the sequence are relevant, we develop a robust model-selection criterion that maximizes the conditional dependence of the response - the number of fragments covering a location in the genome - on the predictors from each model. Greater dependence of the responses on the predictor means that a model based on that predictor can better correct for non-uniformity. The chosen model then corrects for the unwanted coverage biases. Moreover, we show a correspondence between the selected model for each data set and a parameter in the experimental protocol: the average length of the sequenced fragment.
Chapters 3 and 4: Encoding models for the visual cortex (neuro-science)

In chapters 3 and 4, we develop and study encoding models for natural image stimuli. Encoding models are computational models that try to quantitatively describe and predict the relation between a natural visual stimulus and the neural brain activity it evokes. They help evaluate competing theories regarding the function of cortical areas or individual neurons, by measuring how well they can predict the data. Once the accuracy of prediction is established, the model can be analyzed with the hope to better understand the representation of images in the brain.

The visual cortex is composed of a hierarchy of functionally homogeneous areas. For each area, our goal is to find and characterize a limited set of representations of the input images, a feature-set, that could be fit to predict well the evoked neural activity of many neurons in that area. The models and methods described in these chapters owe much to the extensive collaboration with researchers in the Gallant lab, who recorded all data sets and have set much of the ground-work assumed in these chapters.

In Chapter 3 we evaluate the role of the signal-to-noise, contrasted with modeling error, in the prediction accuracy results of encoding models in visual areas V1 and V4. This work is adapted from Benjamini and Yu (2012). Explainable variance, the proportion of signal variance in total variance, helps neuroscientists assess how much the controlled stimulus drives the observed response. We develop a new method to estimate explainable variance, which is tailored for the repeated measure design often used in functional MRI experiments. In particular, we replace classical methods that assume iid noise, an assumption that is not supported by the data, with a new method that only assumes stationarity of the noise.

When applied simultaneously to many responses from an experimental data set, this analysis provides striking patterns within and between functional areas in the cortex. For voxels within the visual area V1, the optimal and observed prediction metrics display a strong linear dependence. This suggests that subsets of the features, based on Gabor filters, can uniformly well fit all voxels within area V1. Voxels within the higher visual area V4, however, display lower accuracy levels for similar explainable variance levels. This suggests a greater role of modeling error in V4 voxels compared with V1, and gives hope of improving the prediction if a more appropriate feature set would be used.

Following this insight, in Chapter 4 we develop a new feature set that better predicts neural activity in the V4 area than the Gabor representation. The chapter is based on the manuscript Mairal, Benjamini, Willmore, Oliver, Gallant, and Yu (2013). We develop this features set on data composed of 71 neuronal responses to a natural-image sequence, recorded by the Gallant lab using electro-physiology. The chapter follows all stages of the analysis of a complex object-prediction problem: analyzing signal-to-noise in the responses; the development of a relevant feature set; fitting of prediction models; and the analysis of these models to form
hypotheses regarding the underlying system.

Of particular interest is the use of unsupervised feature learning in analyzing experimental data. We learned a feature dictionary based on three known organizing principles of V4: orientation channel inputs, invariance for minor deformation and translation, and efficient coding of natural images. We then fit a regularized lag-effect linear model for each neuron to capture the aggregate impact of the observed image sequence. This innovative model predicted with an accuracy comparable to that achieved with better understood visual areas, such as V1 and V2. Interpretation of the models is challenging due to the complicated structure that includes numerous strongly-correlated feature vector and 71 dense linear coefficient vector, that predict the neurons with variable level of success. We propose several methods to approach this challenge.
Chapter 2

Summarizing and correcting the GC-content bias in high throughput sequencing

Biologists use high-throughput sequencing platforms to estimate the abundance of a genomic region from the number of sequenced fragments originating in that region. The GC-content bias describes an observed artifact of DNA-sequencing, a dependence between the count of fragments covering the region and the number of Gs and Cs in the sequence of the region. This bias can dominate the signal of interest for analyses compares abundances within a genome, such as copy number estimation (DNA-seq). The bias is not consistent between samples, and there is no consensus as to the best methods to remove it or even estimate in a single sample.

In this chapter, we fit and compare prediction models for the GC-content bias for several DNA samples. Underlying the analysis is a decision to treat each base-pair in the genome as a separate example; this allows us to compare many representations of GC-content and other sequence based descriptors, identifying which representation best predicts the variation in the coverage. To deal with the huge volumes of data, we develop a new conditional dependence measure that efficiently compares different models. Selection of the model that maximizes this dependence reveals a recurring pattern: it is the GC content of the full DNA fragment, not only the sequenced read, that most influences fragment count. This result is validated across different protocols, labs, and fragment sizes.

The prediction models are used both for correcting the bias and for identifying its source. The base-pair level model allows strand-specific bias correction regardless of the downstream smoothing or binning. The corrected coverage estimates that account for biases by using our prediction model are closer to uniform compared with other models. By relating the
2.1 Introduction

Since it was introduced, Illumina Genome Analyzer high-throughput sequencing has become an increasingly popular technology for determining relative abundance of DNA in an assay. In this method, the DNA of interest is fragmented, and one or both ends of the fragment sequenced. These sequenced short reads, or read-pairs, are aligned to a reference genome. Counts of aligned fragments may be used to measure DNA copy number (DNA-seq), protein binding (ChIP-Seq) or expression (in RNA-seq). In many of these assays, researchers would like to compare such fragment-counts between different locations in the genome. It is therefore troubling that the number of reads mapped to a genomic region depends considerably on the sequence itself.

One well-documented (Dohm et al. 2008) dependency is the GC-content bias, that between the proportion of G and C bases in a region and the count of fragments mapped to it. (We use fragment count, instead of read count / read coverage because paired reads identify a fragment). This variability does not reflect the signal of interest, but might confound it. Because GC abundance is heterogeneous across the genome and often correlated with functionality, the GC effect can be hard to tell apart from the true signal. The effect does not decay even for larger bins: large (>2-fold) differences in coverage are common even in 100 kb bins (Chiang et al. 2008). To make matters harder, the effect is not consistent between repeated experiments, or even libraries within the same experiment (see below). Estimating and directly correcting for this effect has become a well-established step in protocol design (Quail et al. 2008), quality control (http://picard.sourceforge.net/index.shtml), and studies (Ivakhno et al. 2010; Yoon et al. 2009) using high throughput sequencing.

Most current correction methods follow a common path. Both fragment counts and GC counts are binned to a bin-size of choice. A curve describing the conditional mean fragment count per GC value is estimated (by binning, Yoon et al. 2009, or assuming smoothness Boeva et al. 2011; Miller et al. 2011). The resulting GC curve determines a predicted count for each bin based on the bin’s GC. These predictions can be used directly to normalize the original signal, or as the rates for a heterogeneous Poisson model. Bin size is arbitrarily set, usually to match down-stream analysis. While these methods remove most of the GC effect, they do not use any prior knowledge about the effect. This is perhaps why key features of
the GC curve, such as its unimodality, have sometimes been overlooked or completely missed in the estimation.

While GC effect is commonly corrected for, until recently studies regarding the nature of this bias have been rare. Dohm et al. (2008) first described the effect of the GC on fragment coverage in the Illumina GA sequencing platform. The effect they found seemed highly linear - fragment coverage increased with GC content, but they sequenced genomes that were GC poor. This is probably why the GC effect is sometimes described as the correlation between GC and coverage (Teytelman et al. 2009). In later high-throughput studies of the human genome, plots of GC-curves usually reflect non-linear curves, but are rarely investigated further than non-parametric fitting. Identifying the source of the bias was also hard, because the composition of the DNA molecule can affect many stages of the protocol. Sequence related biases in the priming (Hansen et al. 2010), size-selection (Quail et al. 2008), PCR (Kozarewa et al. 2009), and probability of sequencing-errors (Nakamura et al. 2011; Aird et al. 2011; Bravo and Irizarry 2010) have all been found. In a recent analysis (Aird et al. 2011), PCR was shown to play the dominant role in the stages before the sequencing. While sequencing protocols have partially evolved to accommodate this new understanding (Kozarewa et al. 2009; Aird et al. 2011), estimation and correction methods have not.

From a technical point of view, the above sources of bias cluster according to the location and scale of GC thought to be driving the non-uniformity in the counts. Locally, GC counts could be associated with the stability of the DNA, and thus modify the probability of a fragmentation-point occurring in the genome, leading to a fragmentation model. The GC content could primarily modify the base-sequencing process; we call this the read model, suggesting that the GC of the forward read (in the single-end) or both reads (in the paired-end case) best explain fragment count. Full-fragment models assume the GC of the whole fragment determines which fragments are selected or amplified. Finally, global models refer to GC effects on scales larger than the fragment length, e.g., through an association with some higher-order structure of the DNA. These loosely defined models can be realized statistically by counting the GC in a suitable region and comparing that to fragment coverage. While the differences between the above models might seem small, they are sometimes considerable (see below). Note that any GC-bias removal strategy implicitly chooses a GC-bias model when it uses GC in some region to correct for the effect.

In this work, we take a descriptive approach to investigating the common structures found in GC-curves in DNA-seq. We study the effect of GC on fragment count in many DNA-sequencing copy-number assays for (both normal and tumor) high-coverage human genomes, taken from multiple labs. Copy-number for normal genomes should rarely change, and so observed variability in fragment count can almost always be attributed to technical effects rather than biological signal. We use a single position model to estimate the effect of GC on the fragment counts, and seek a parsimonious description for this family of curves. (The same model underlies the correction method BEADS, see Cheung et al. 2011). Such a description
has two main advantages: it allows more accurate estimation of the GC curve by highlighting an appropriate set of parameters; and it provides important empirical evidence regarding the experimental stages that may cause or modify this effect.

The data we analyze suggests that to a large extent, the dependence between count and GC originates from a biased representation of possible DNA-fragments, with both high GC and high AT fragments being underrepresented. This global structure of the GC dependence is consistent, but the exact shape varies considerably across samples, even matched samples. We describe a parsimonious model for the GC effect, and show it suffices to predict the GC effect on fragment coverage on all scales, all chromosomes and for both strands. This prediction is better than generic fitting approaches currently used, as illustrated on DNA-seq with and without copy-number events. Our model produces single base pair prediction, allowing optimal correction regardless of the required downstream smoothing. Finally, this provides empirical evidence strengthening the hypothesis that PCR is the most important cause for GC bias.

2.2 Materials and Methods

2.2.1 Mapped read files

The main data set we use consists of two samples of DNA from an ovarian cancer patient: one sample from the tumor and another normal sample (from white blood cells). Each of these DNA samples was turned into two separate fragment libraries, differing in fragment length distribution. Fragments were sequenced on both ends - 75 base-pair reads on each end - according to standard Illumina procedures. Each fragment was then mapped back to the human reference genome, based on the 5’ read. These sequenced read pairs were mapped to a reference using BWA (version-0.4.9, Li and Durbin 2009). Human genomes were mapped to NCBI build 36 version 3 (ftp.ncbi.nih.gov/genomes). The data available at the NCBI Sequence Read Archive under accession numbers SRX011739 (tumor) and SRX011777 (normal). Unless otherwise mentioned, all plots are from chromosome 1 (chr1) of normal Lib1. Additionally we analyzed another sample from a breast tumor cell line, healthy genome libraries generated under optimized protocols (Supplementary Figures S5, S6), and data from a ChIP sequencing experiment of Arabidopsis (Supplementary Figures S7, S8). For details see Table 2.1.

For each library, this procedure resulted in a list of fragments. Each fragment can be described by the location of its 5’ end (chromosome, location and strand), and its length. Length was inferred from the mapping of the 3’ end, based on the paired-end alignment of BWA. Only those pairs in which the 5’ read was uniquely mapped were kept (flag XT:A:U of
BWA), because allocation of reads mapped to multiple locations is very sensitive to the comprehensiveness of the reference. However, we did not discard a pair when the 3’ was not mapped; thus for most of this analysis, the fragment set is similar to that obtained from single-end data. Only where fragment length is explicitly discussed did we remove fragments when the 3’ was not mapped (∼1% of fragments). No additional quality filtering was applied.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Date</th>
<th>Institute</th>
<th>Fragments (bp ± SD)</th>
<th>Reads (bp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCGA-13-0723</td>
<td>Matched</td>
<td>4/09</td>
<td>Wash U</td>
<td>154 ± 17</td>
<td>75</td>
</tr>
<tr>
<td>Normal Lib1</td>
<td>Matched</td>
<td>4/09</td>
<td>Wash U</td>
<td>284 ± 38</td>
<td>75</td>
</tr>
<tr>
<td>Normal Lib2</td>
<td>Matched</td>
<td>4/09</td>
<td>Wash U</td>
<td>173 ± 22</td>
<td>75</td>
</tr>
<tr>
<td>Tumor Lib1</td>
<td>Ovarian T</td>
<td>4/09</td>
<td>Wash U</td>
<td>293 ± 31</td>
<td>75</td>
</tr>
<tr>
<td>Tumor Lib2</td>
<td>Ovarian T</td>
<td>4/09</td>
<td>Wash U</td>
<td>173 ± 22</td>
<td>75</td>
</tr>
<tr>
<td>HCC1569</td>
<td>Breast T</td>
<td>1/09</td>
<td>UC Berkeley</td>
<td>507 ± 40</td>
<td>45</td>
</tr>
<tr>
<td>SRX040660</td>
<td>Matched</td>
<td>9/10</td>
<td>Broad</td>
<td>172 ± 19</td>
<td>101</td>
</tr>
<tr>
<td>SRX040661</td>
<td>Matched</td>
<td>9/10</td>
<td>Broad</td>
<td>173 ± 19</td>
<td>101</td>
</tr>
<tr>
<td>ChIP Rep 1</td>
<td>Arab.</td>
<td>7/09</td>
<td>UC Berkeley</td>
<td>100†</td>
<td>36</td>
</tr>
<tr>
<td>ChIP Rep 2</td>
<td>Arab.</td>
<td>7/09</td>
<td>UC Berkeley</td>
<td>100†</td>
<td>36</td>
</tr>
</tbody>
</table>

Table 2.1: List of data sets analyzed in this chapter

2.2.2 Loess model

Previous analyses of GC focused on the relation between fragment count and GC composition for particular bin sizes (see for example Boeva et al. 2011; Kuan et al. 2009). We call this the loess model, and describe shortly how we reproduce such estimation. For a given bin (interval of the genome), GC is the fraction of G and C bases in that bin according to the reference genome. Bin counts are the total number of (forward) fragments with 5’-end inside the bin. To account for uniqueness of sequences, a mappability measure is calculated for each position (base pair) in the bin. A location is called mappable if the k-mer of the reference genome starting at the location is not perfectly repeated at any other location in the genome, where k is the read length (checked by a Python script using the Bowtie mapper Langmead et al. 2009). All subsequent analysis is done in R (R-Development-Core-Team 2010). The GC-bias curve is determined by loess regression of count by GC (using the loess R package) on a random sample of 10,000 high mappability (> 0.9) bins. The smoothness parameter for the loess should be tuned to produce curves that are smooth but still capture the main trend in the data. We use 0.3 as the default value. The estimated rates for GC values with too few data points are set to 0.
2.2. MATERIALS AND METHODS

2.2.3 Single position models

We call single position models those models that estimate the mean fragment count (the rate) for individual locations rather than bins. We consider a family of such models linking fragment count to GC, where the expected count of fragments starting (5’ end) at \( x \) depends on the GC count in a window that starts \( a \) bp from \( x \). Figure 1 illustrates the case in which GC is computed from a 4 bp window starting at the fragment 5’ end. Each such model can be characterized by the shift \( a \) and the length \( l \) of its “driving” GC window. \( W_{a,l} \) denotes the model in which the fragment count starting at \( x \) (5’ end) depends on the GC between \( x + a \) and \( x + a + l \). (We will use \( GC(x + a, l) \) for the GC count of the \( l \) bp window starting at \( x + a \).) For example, in \( W_{0,r} \) the fragment count is determined by the GC of the first \( r \) bp of the fragment. The model \( W_{a,l} \) has \( l + 1 \) rate parameters, \( \lambda_0...\lambda_l \), corresponding to windows with \( gc = 0, ..., l \).

Figure 2.1: Single position model estimation. (A) Mappable positions along the genome are randomly sampled (⇓); (B) These positions are stratified by the GC count in the corresponding sliding window (here \( a = 0, l = 4 \)); (C) The number of fragments (→) with 5’ end (○) in sampled locations are counted; (D) Mean fragment rates for each stratum are estimated, taking the ratio between fragment count and positions in the stratum. These form the GC curve (here the curve for \( a = 75, l = 50 \) from Figure 2.4, Panel C).

\[
W_{a,l}
\]
2.2. MATERIALS AND METHODS

The following is a description of our method of estimating the parameters for a single position model $W_{a,l}$. A large ($n \approx 10$ million) random sample of mappable locations is taken from the genome. Large genomic regions with either zero fragment-counts or with counts that are extremely high (> 0.99 quantile + median) were removed from the sample. The sample is partitioned (stratified) according to the GC of the reference genome: if $gc = GC(x + a, l)$ then position $x$ is assigned to stratum $S_{gc}$. Let $N_{gc}$ denote the number of sample positions assigned to $S_{gc}$. Note that the assignment to strata depends only on properties of the model and the reference genome, not on the sequencing data.

Next, for every value of $gc$, we count the total number $F_{gc}$ of fragments starting (5' end) at the $x$’s in $S_{gc}$. We estimate $\lambda_{gc}$ by taking the ratio

$$\hat{\lambda}_{gc} = \frac{F_{gc}}{N_{gc}}$$

The random sample is taken over (potential) positions in reference genome, not fragments. The estimated fragment rates implicitly account for the total number $F$ of mapped fragments in the sample ($F = \sum_{gc=0}^{l} N_{gc}\hat{\lambda}_{gc}$). For large windows, we expect many strata to be small. Strata are then pooled together (constant jumps of 3 or 6 are used). The parameters that were skipped are then estimated by interpolation using loess regression (smoothness 0.2).

Comparing models

An estimated model $W_{a,l}$ (i.e. each choice of GC-window) can be used to generate predicted counts for any genomic region. Models can be compared based on the quality of their corrections (see below). However, this is very inefficient, and we consider a simpler surrogate measure that allows comparison of many different models regardless of window size. We use the normalized total variation distance (TV, e.g. Durrett (2010)) between the stratified estimated rates ($W_{a,l}$) and a uniform rate ($U$, equal to the global mean rate in our sample $\hat{\lambda} = F/n$)

$$TV(W_{a,l}, U) = \frac{1}{2\hat{\lambda}} \sum_{gc=0}^{l} \frac{N_{gc}}{n} |\hat{\lambda}_{gc} - \hat{\lambda}|.$$ 

The above (TV) score is a weighted $L_1$ distance from the global mean, divided by $2\hat{\lambda}$ (so it will be between 0 and 1). In other words, it is the total variation distance between the empirical distribution for a single fragment (under specific GC categories) and a uniform distribution. Thus, it measures the proportion of fragments influenced by the stratification, and is comparable across data sets. We look for high TV, meaning counts are strongly dependent on GC under a particular stratification. This could indicate that correcting for such a model would best correct for the GC dependence.
2.2. MATERIALS AND METHODS

Fragment length models

To measure the effect of fragment lengths, a separate single position model is fit for fragments of each length. $W_{a,l}^s$ accounts for the fragments of length $s$ only. The locations in the sample are still partitioned according to $gc$, but instead of counting all fragments starting at $x$’s in $S_{gc}$, only fragments of length $s$ are counted ($F_{gc}^s$). Rates are estimated as before,

$$\hat{\lambda}_{gc}^s = F_{gc}^s/N_{gc}.$$  

For the fragment length model, we would like to predict the count of fragments using the GC in the fragment after removing a few bp from each end to reduce the impact of the local biases. Hence, the GC-window size $l$ becomes $s - a - m$. The model $W_{a,s-a-m}^s$ is then determined by $a$ the shift from the fragment 5’ end, and $m$ the margin from the 3’ end. (Note that if $l$ had been instead fixed for any $s$, the set $\{W_{a,l}^s\}$ would be a refinement of $W_{a,l}$, with $F_{gc} = \sum_s F_{gc}^s$ and $\hat{\lambda}_{gc} = \sum_s \hat{\lambda}_{gc}^s$. This is not the case here, because the GC-window grows with the fragment length.) The parameters of this model stand for rates for each combination of fragment length and GC. The rate surface is smoothed using a 2D Gaussian kernel ($\theta = 0.7$ for estimation, $\theta = 1.8$ for visualization - Figure 2.5).

Predicted rates

The prediction of mean fragment count $\mu_x$ for genomic position $x$ using $W_{a,l}$ is

$$\hat{\mu}_x = \begin{cases} \hat{\lambda}_{GC(x+a,l)} & \text{if } x \text{ is uniquely mappable} \\ 0 & \text{otherwise.} \end{cases}$$

In essence we are smoothing the observed fragment counts using $W_{a,l}$. That is, we are estimating the number of fragments at $x$ under the model $W_{a,l}$, by the average of all such numbers found in $x$’s with the same value of $GC(x + a, l)$ (in our sample of mappable locations). Therefore, when we wish to remove - correct for - the GC bias, as assessed by $W_{a,l}$ at $x$, we would divide the observed number of fragments emanating from any $x$ by $\mu_x$. In practice, we rarely use the final correction at the single bp resolution, but usually after aggregating into bins, see below.

For the fragment length model, $\{W_{a,s-a-m}^s\}_s$, the predicted count of fragments from all lengths (if $x$ is mappable) is the sum of predictions for each length

$$\hat{\mu}_x = \begin{cases} c \cdot \sum_s \hat{\lambda}_{GC(x+a,s-m)}^s & \text{if } x \text{ is uniquely mappable} \\ 0 & \text{otherwise.} \end{cases}$$
with \( c \) a scale factor to equalize predicted and observed total fragment counts (based on the fraction of fragments with unknown length). Finally, the mappability model uses the global mean (\( \hat{\lambda} \)) as the predictor for each mappable position

\[
\hat{\mu}_x = \begin{cases} 
\hat{\lambda} & \text{if } x \text{ is uniquely mappable} \\
0 & \text{otherwise.}
\end{cases}
\]

Fragment counts in bins are additive, so for any bin \( b \) the predictor is \( \hat{\mu}_b = \sum_{x \in b} \hat{\mu}_x \). The reverse strand follows similar fragment rates when the GC window direction is reversed (see below). To show this, we estimate the strands separately except where otherwise stated.

### 2.2.4 Evaluation

We evaluate the success of a model by comparing its predictions (\( \hat{\mu} \)) to the vector of observed fragment counts (\( F \)). For robust evaluation, we measure the mean (average) absolute deviation (MAD) between predicted and observed counts. Let \( B \) be the set of bins, and \( F_b \) the count of fragments for which 5' end is inside of bin \( b \).

\[
MAD(F, \hat{\mu}) = \text{avg}_{b \in B} |F_b - \hat{\mu}_b|.
\]

For visualization, observed counts are normalized by their respective predicted values, creating an implicit measure of copy number (CN). That is, we plot

\[
CN_b(F_b, \hat{\mu}_b) = \frac{F_b + \epsilon}{\hat{\mu}_b + \epsilon},
\]

where \( \epsilon = 0.1 \) stabilizes this estimate when the predicted number of fragments is small. On non-tumor data, we expect these values to be concentrated around 1, and the spread should indicate the quality of the predictions.

### Comparisons with Poisson variation

Let \( F_b \) be the fragment count in bin \( b \) as before, \( \mu_b \) the expected value of \( F_b \) assuming a Poisson distribution, and \( \hat{\mu}_b \) an estimate of \( \mu_b \) under some model \( W_{a,l} \). We compare the variation of \( F_b \) around the predicted mean \( \hat{\mu}_b \) to the variation expected under a Poisson distribution. Then the residual variance (RV) is

\[
RV = \frac{1}{|B|} \sum_{b \in B} (F_b - \hat{\mu}_b)^2.
\]
Under the heterogeneous Poisson, the residual variance is composed of a bias term, a term for estimation error, and Poisson variance as following

$$RV \approx \frac{1}{|B|} \sum (\mathbb{E}[\hat{\mu}_b] - \mu_b)^2 + \frac{1}{|B|} \sum (\hat{\mu}_b - \mathbb{E}[\hat{\mu}_b])^2$$

$$+ \frac{1}{|B|} \sum (F_b - \mu_b)^2$$

where the first quantity is a bias term, corresponding to the goodness of fit of the model; the second is an estimation error term due to fitting the model, and should be relatively small; and the last is a measure of the pure Poisson variance, and whose value should be about the average of the $\mu_b$.

Thus residual variance measures how well a model captures the rate parameters, and cannot be less than the pure Poisson variance. We compare the residual variance of the mappability model (MR), and the residual variance of the fragment GC model (GR), to an independent estimate of the pure Poisson variance ($\bar{F}$). Extremely high counts ($F_b > 0.99$ quantile + median($F$)) were removed from the computation.

Although we remove extreme high counts, variances can still be influenced by a relatively small set of bins with high counts. We would like to compare residual variation of the different models in a way that is more robust to those. Following (Bentley et al. 2008), we look at the empirical quantile curves. For a given model, we grouped counts from 1 kb bins according to their predictions ($\hat{\mu}_b$’s). We computed for each group the observed 0.1 and 0.9 quantiles. Plotted against the estimated rate of the group, each quantile level forms a curve. The distance between the curves reflects the variation around predicted means. When this distance is considerably larger than that of the Poisson, this indicates that different rates were assigned to the same predicted value, meaning that much variation remains unexplained by the model.

### 2.2.5 Software and data availability

GCcorrect is an R package implementing exploratory analysis, estimation and correction methods for GC content effects, and is available for download from http://www.stat.berkeley.edu/~yuvalb.
2.3 Results

2.3.1 Bin Counts

The GC effect for human genomes is largely unimodal. In AT rich regions, coverage increases with increasing GC. In GC rich regions, coverage decreases with increasing GC. The peak coverage can be different for different data sets (and bin sizes), but is usually located between 0.4 to 0.55 GC. That 10 kb bins with GC > 0.5 are rare in the human genome is perhaps the reason for calling GC effect ‘linear’. This unimodal relation can be seen at almost any scale, from 50 bp to above 100 kb. While in the AT rich region the increase in coverage is quite linear with GC, it is less linear (and more variable) in GC rich regions.

![Figure 2.2: GC curves (10 kb bins)](image)

Observing fragment counts and loess lines plotted against GC of (A) two libraries from the same normal sample, and (B) the tumor library (red) with its matched normal sample library (blue). Counts and curves of all libraries are scaled to fit median counts of normal library 1. Bins were randomly sampled from chromosome 1, and counts include fragments from both strands.

The curves of difference samples are all unimodal, but not the same: The slope, location of mode, and variance around the unimodal curves vary considerably between samples. Indeed, variability between curves is found not only between labs or protocols, but also between tumor and normal sample pairs and between different libraries based on the same starting DNA. In Figure 2.2 (A), we compare the GC curves of two libraries prepared from the same normal genome. The curves are not aligned: for GC poor bins fragment counts of library 1 (dark blue) are higher compared to library 2 (aqua green), while in GC rich bins fragment
counts of library 1 are lower. Moreover, the GC curve (B) of normal library 1 (blue) does not follow the curve of tumor library 1 (red), displaying both different slopes and different peak locations (B). The curve for the tumor peaks at a GC of 0.55, but for the normal library 1 it peaks at 0.48 (and library 2 at 0.5). (That tumor has only a single band reflects that there were no large copy-number events in chromosome 1; this is not true in general, see for example Figure 2.9) This makes a case for the importance of single sample normalizations (and library specific normalizations). However, different regions in the same normal genome (Supplementary Figure S1) do have similar GC curves. Also, different lanes of the same library (on a single flow cell) display the same curve (not shown).

The choice of bin size should not matter much if the bias is linear. However, sampling a unimodal curve at the wrong scale will normally increase the variance. We therefore compared the absolute deviance from the curve at different bin sizes (Table 2.2). The predictions were aggregated to 10 kb (regardless of estimation bin size). The results improve as bin size decreases. This is true, for both libraries, until we approach bin sizes of the order of the fragment length (300 bp for library 2, 200 bp for library 1). Counts of library 2 were scaled by median fragment rate to match library 1.

<table>
<thead>
<tr>
<th>Loess bin size (kb)</th>
<th>10</th>
<th>5</th>
<th>2</th>
<th>1</th>
<th>0.5</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Lib1</td>
<td>49.1</td>
<td>47.8</td>
<td>45.1</td>
<td>43.4</td>
<td>43.4</td>
<td>52.2</td>
</tr>
<tr>
<td>Normal Lib2</td>
<td>54.8</td>
<td>52.0</td>
<td>47.5</td>
<td>45.7</td>
<td>49.7</td>
<td>87.7</td>
</tr>
</tbody>
</table>

*Table 2.2: Prediction error (MAD) of loess model for different resolutions.* Error measured by mean absolute deviation around the predicted rates. The best predictions (minimal error) for each library are in bold. Rates were estimated using loess at the specified resolution, and then predictions were aggregated into 10 kb bins.

Indeed, we cannot expect reducing bin sizes to work for such small scales. On scales comparable to fragment sizes the bin-edge effects become substantial. Each of the different models for the GC effect (fragmentation, reads, or full fragments) should imply a different correction strategy. Moreover, small bins have few reads / fragments, and so measuring variability around the mean becomes harder. Instead of binning, single position models (see Figure 2.1) are introduced to measure GC effects in these smaller scales.

### 2.3.2 Single position models

Single position models allow us to compare different possible GC windows, estimate the effects for each, and compare their TV scores. First we compare TV scores of GC windows starting at the 5’ end ($a = 0$) of a location (but having different lengths). We would expect to see the strongest effect either after a few bp (fragmentation effect), after 30-75 bp (read effect) or at the fragment lengths (full-fragment effect).
2.3. RESULTS

Figure 2.3: **Single position models.** (A) The top curves represent TV scores for GC windows of different lengths, all beginning at 0 (a=0). The horizontal bars on the bottom mark the median fragment lengths (and 0.05, 0.95 quantiles). For each library, the strongest GC windows are those that encompass the full fragment. For library 1, we mark the optimal model ($W_{0,180}$), and show its resulting GC curve on the right panel (B). (We actually show $W_{2,176}$, removing 2 bp from each side of the fragment). The GC curve measures the fragment rate given the fraction of GC in the window. Vertical lines (blue) represent 1 std-dev. For comparison we plot the distribution of GC (dotted line) in our sample from chromosome 1 (scaled).

For both libraries, the full-fragment model achieves the highest TV score. In Figure 2.3 (A), the two curves represent TV scores of the two libraries from the normal sample. The horizontal bars on the bottom mark the median (and 0.05, 0.95 quantile) fragment sizes for the two libraries. TV-scores for both libraries increase as the window size increases, with the strongest effects for windows almost matching the median fragment length: strongest effect for window of length 180 ($W_{0,180}$) for library 1 (median length = 174), and length 295 ($W_{0,295}$) for library 2 (median length = 293). For windows longer than that, the scores decrease.

The GC-curve that is estimated from the window $W_{2,176}$ is extremely sharp, see Figure 2.3 (this is $W_{0,180}$ after removing 2 bp on each end). In fact, strong unimodality can be seen on even smaller scales. Smaller windows ($l = 50$ bp) allow us to contrast a GC window that overlaps the read with a GC window that does not ($W_{0,50}$ vs $W_{75,50}$). (Figure 2.4, B and C). The GC-effect estimated from both windows has a unimodal shape, but the curve of the window overlapping the read is not as sharp as that of the window from the fragment center. If read-composition were driving the GC-effect we would expect the first window to generate the sharper curve. That this is not the case, may imply that the GC effect is not driven by base calling or sequencing effects, but by the composition of the full fragment. (Rather, the
2.3. RESULTS

Figure 2.4: Different lags (A) GC curve of the window before the fragment - $W_{-50,50}$, (B) within the read - $W_{0,50}$, and (C) in the fragment center, not overlapping the read - $W_{75,50}$. (D) A plot of TV scores for 50 bp sliding windows ($W_{a,50}$). The x-axis marks $a$, the location of the window 5’ end relative to 5’ end of the fragment. On the bottom we mark a fragment and its reads in relation to the GC windows from the top panels.

sharper curves in the center imply a second weak bias near fragment ends, see below). For contrast, panel A shows the GC curve estimated from the 50 bp located just outside the fragment ($W_{-50,50}$). The curve is not unimodal, and has a noticeably lower TV score.

To further illuminate this (Panel D), we compute TV scores for other 50 base-pair GC windows with different lags. The TV curve traces the shape of the fragments: It ascends sharply for windows completely within the fragment, and then dips considerably for windows outside the 3’ end. The line is mostly symmetric around half the median-fragment length, decreasing as the windows extend over the 3’ ends of fragments. In fact, enumerating over many positions $a$ and lengths $l$, the strongest windows are those overlapping most of the fragment but excluding the fragment ends. (Note that the 5’ end is perfectly aligned, the 3’ is not, due to varying fragment lengths). The TV scores decay outside the fragment, but still reflect some GC dependence due to large scale correlations in GC composition.

In Supplementary Figure S2 we contrast the TV plots generated from the forward strand
with TV plots of the reverse strand. While the reads have exactly the same location (no matter the strand), the forward strand fragments extend to the 3’ end of the read while reverse strand fragments extend to the 5’ end. The TV score lines trace these shapes. After the proper inversion and shift, both GC curves estimated on the reverse strand and their TV scores match those from the forward strand.

### 2.3.3 Effect of fragment length

Within a library, we find that the length of fragments influences the shape of the GC curve. If GC depends on fragments and not reads, the GC is a quotient of two fragment parameters: the number of G and C bases, and the length of the fragment. We might expect the two parameters to interact to determine the rate of fragments. This is indeed the case. Within a single library, GC curves estimated on longer fragments peak at higher GC’s.

---

**Figure 2.5:** Fragment rate by length and GC. (A) A heat map describes rates for each (GC, length) pair. Each dotted line represents a single length. In (B), GC curves for fragments of specific lengths are drawn (corresponding to the dotted lines in (A)). Blue / dark curves represent shorter fragments than red / bright. Here x-axis is the fraction of GC. All fragment length models here have a margin of 2 from both fragment ends ($a = 2$, $m = 2$).

Figure 2.5 (A) displays a surface describing fragment rates for all (GC, length) pairs. We use the GC count of the full fragment excluding the first 2 bp on each end, corresponding to $W_{2,s-4}^2$. Each horizontal cut of this surface represents a GC curve for fragments of a specific length.
2.3. RESULTS

Models restricted to long fragments (top of Panel A) tend to reach highest rates at higher GC counts (right). The shift toward high GC in longer curves persists in the rescaled curves (Panel B). The curves displayed here are represented by the dotted lines on (Panel A), but this time rescaled so that the x-axis is the fraction GC, not the count. We have seen similar patterns of GC-length interactions in other data sets from different sequencing centers, though not all.

2.3.4 Local biases near fragment ends

While the unimodal effect is the strongest inhomogeneity in coverage, it is not the only one. We will discuss two (perhaps partially related) effects that are found near the fragment ends, and argue they are not driving the GC effects at larger bin sizes.

The first of these is a preference of AT near the fragment ends. Note that the GC curve based on a window just 5’ to the fragment (Figure 2.4, panel A) reveals a second mode of AT rich windows (in addition to the mode at 0.5 GC). Traces of this mode can also be seen in Panel B overlapping the read. The TV score when stratifying by this window are indeed lower (compared to the center of the fragment), reflecting the conflicting effects. This phenomena is strongest for 20-30 bp surrounding both the 5’ and 3’ end.

A second bias is in the composition of the few bp around the fragment ends. It has been described before in RNA-seq [Hansen et al. 2010]. The relative frequency of nucleotides follows a position-specific pattern roughly starting 4 bases before fragment and ending 8-9 bases inside it (see Figure 2.6, Left). We call this the fragmentation effect. Note that G and C are differently preferred, and so is A compared to T. Complementary effects can be seen on the 3’ end of the fragment, for a fixed fragment size. The fragment-GC effect described before can also be seen - the small preference of G and C between 20 and 200 (reflecting fragment sizes). Rates stratified by dinucleotide counts are significantly different than singletons. In particular, the dinucleotide on which fragment rates depend the most is the pair surrounding the fragment end (the breakpoint), shown on the right. Fragments are much more likely to start within a CpG dinucleotide, than any other dinucleotide.

2.3.5 Aggregating effects and corrections

Local effects captured by the fragment model drive the GC curves found at larger scales. In Figure 2.7, panels (A)-(C) compare GC to predicted and observed bin counts at various bin sizes. For all three bin-sizes, the predicted counts (black) trace the observed loess line (blue), but also capture some of the variability around the curve.
2.3. RESULTS

Figure 2.6: **Fragmentation effect.** Left: Relative abundance of nucleotides at fixed positions relative to fragment 5’ end. A horizontal dotted line marks the relative abundance of the base at mappable positions. Right: Fragment rates when stratifying by the dinucleotide (−1,0). Dinucleotide counts overlapping the fragment 5’ end.

In contrast, models based on smaller portion of the fragment do not trace the observed curves. Panel D shows the estimates from the read ($W_{0.75}$). The predictions are too high for GC rich or GC poor bins, and too low for intermediate GC bins. Similarly, the **two end model** (E), using GC (30 bp) from both ends of the fragment, produces unimodal predictions which are not sharp enough to capture the observed shape. Prediction based on the **fragmentation model** (F) does not produce sharp contrasts or unimodality. The methods of correction used for (E) and (F) are described in detail in Supplementary methods.

Correction based on the fragment and fragment-length models remove most GC dependent fragment count variation. Predicted counts based on the fragment model are more accurate than predictions from the optimal loess model (MAD = 9.5 for fragment model, compared to 10.8 for loess model on 1 kb bins). The same holds for all bin-sizes. Adding fragment-length into this model slightly improves the prediction quality (MAD=9.1). Because adding length did not change the results greatly, we use the more parsimonious model for the rest of this work.
2.3. RESULTS

Figure 2.7: **Aggregation of single location estimates.** (A-C) Estimates based on the fragment GC curve (black) trace similar paths as loess (cyan) estimated on observed counts (blue) on multiple scales. (D-F) Estimates based on alternative models compared to observed counts on 1 kb bins. (D) **Read model**, predictions based on GC of the 5’ read only ($W_{0.75}$); (E) **Two-ended model** uses GC (30 bp) from both ends of the fragments; (F) **Fragmentation model** based on location specific composition around the 5’ end. See Supplementary methods for details on how models for (E) and (F) were defined and estimated.

We visualize the correction in a region of chromosome 1 which has no copy number (CN) changes (Figure 2.8). On the left, uncorrected (but scaled) 1 kb bin counts display large low-frequency variations, which can be mistaken for CN events. The fragment model removes these variations better than the loess model. In (B), a histogram of corrected counts shows that the fragment correction produces tighter distribution of scaled counts around 1 compared to the loess model.

A similar correction on the tumor data reveals a hidden CN (both libraries, forward strand) in Figure 2.9. GC-curves (for both the loess and fragment models) were estimated from chromosome 1, and corrected counts for a CN gain on chromosome 2 are shown. The CN gain is hidden in the uncorrected data due to low-frequency count variation driven by GC-content. Both the fragment model correction and the loess correction reveal the CN gain. The fragment correction provides better separation between bands (see histograms in B).
2.3. RESULTS

Figure 2.8: Corrected counts of normal sample. (A) Counts in 1 kb bins not corrected for GC (top), corrected by loess (center), and corrected by fragment model (bottom), positions 58,000-62,000 kb of chromosome 1 (chr1). (B) Histograms of the corrected counts (random sample of 1 kb bins in chr1). Each point represents counts from both libraries (forward strand).

Also, it successfully corrects for different binning resolutions (see Supplementary Figure S3). Note that chromosome 1 was used for GC estimation because it does not seem to have large CN changes (as seen in Figure 2.2).

2.3.6 Poisson and other variation

The estimated GC-effect and mappability explain most the variation in the fragment coverage of the normal genome (though not all of it). In Table 2.3 we compute the residual variance after removing the GC-effect in 1 kb bins. The GC model removes most of the variability in the binned counts, much more so than corrections based only on mappability. The residual variance of the fragment model is considerably smaller than that of the loess model. It is still larger than Poisson, though small areas with extremely high coverage cause most of this extra variance.
2.3. RESULTS

Figure 2.9: Copy number (CN) gain from tumor sample Counts and corrected counts at position 29,000 kb on chromosome 2. (A) Unnormalized counts at 1 kb bins (top), corrected by loess (center), and corrected by fragment model (bottom). GC curves estimated on chromosome 1 (which has no large CN changes). (B) Histogram of normalized counts at 28-30 mb (underlined on left plots).

<table>
<thead>
<tr>
<th>Method</th>
<th>Total</th>
<th>MR</th>
<th>GR</th>
<th>1-GR/MR</th>
<th>P</th>
<th>GR/P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loess</td>
<td>909</td>
<td>464</td>
<td>177</td>
<td>0.61</td>
<td>59</td>
<td>3</td>
</tr>
<tr>
<td>Fragment</td>
<td>909</td>
<td>464</td>
<td>137</td>
<td>0.7</td>
<td>59</td>
<td>2.33</td>
</tr>
</tbody>
</table>

Table 2.3: Residual variance from different models. Residual variance of GC models (GR), compared to residual variance from mappability model (MR) and to the expected variance of a heterogenous Poisson (P). Also displayed are the proportion of residual variance (after mappability correction) explained by GC (1-GR/MR); and the ratio between GC residuals and the expected Poisson variance (GR/P). Computed on 1 kb bins from normal sample (forward strand, library 1), after removing outlier bins.

For a comparison more robust to these high-coverage regions, we compare quantiles rather than variances. In Figure 2.10, we compare the 0.1 and 0.9 quantiles of observed counts grouped by the estimated fragment rates of different models (see Methods). The variation in bins with very low observed counts is largely explained by mappability. However, mappability cannot
2.3. RESULTS

explain variation of higher counts, and the spread between the quantiles is approximately double that of the Poisson. Models taking GC-content into account produce much tighter spreads. The fragment-length model (the green curve) consistently leaves less variation around the estimated rates than the loess model (blue).

Figure 2.10: **Comparison to Poisson variation.** 0.1 and 0.9 quantiles of observed counts grouped by estimated rates. Models that predict better will have narrower vertical spreads. Variation around the mean of the fragment model (green), the loess (blue) and mappability (black) are compared to variation around a Poisson (red).

2.3.7 Additional data sets

In the above analysis we described a single tumor-normal pair produced by a single lab, but our results are general to many examined samples from multiple labs. In Figure 2.11 we show 4 descriptive plots from a different data set (based on HCC1569 cell line, see Table 2.1 for details). The GC has a strong effect on fragment counts, and this relation is unimodal (A). The highest TV score is for a window of approximately fragment length (B), resulting in a sharp GC curve (shown in C) which predict the GC trends (D). A distinct difference is the lack of length-dependence of the fragments (not shown). The AT preference near fragment
2.4 Discussion

Large biases in fragment counts related to the GC composition of regions were found in the data sets we examined. These observed effects have a recurring unimodal shape, but varied considerably between different samples.

We have shown that this GC effect is mostly driven by the GC composition of the full fragment. Conditioning on the GC of the fragments captures the strongest bias, and removing this effect provides the best correction, compared to alternative GC windows. When single bp predictions based on the fragment composition are aggregated, the results trace the observed ends is also missing, further evidence that it is not the major source of the GC bias. Two additional sets of data are shown in the Supplementary Data.

Figure 2.11: GC plots for Dataset 2. (A) GC effect for 10 kb (chromosome 1). (B) TV scores for GC windows of different lengths with a=0 (comparable to Figure 2.3). (C) GC curve at fragment model \( W_{2,500} \). (D) Observed (blue) and predicted (black) counts against GC for 10 kb bins (chromosome 2).
GC dependence. This cannot be said about local effects that take only the reads into account. This conclusion holds for various data sets, with different fragment-length composition, read lengths, and GC-effect shapes.

That the GC curve is unimodal is key to this analysis. In all data sets shown, the rate of GC poor or GC rich fragments is significantly lower than average, in many cases zero. Unimodality was overlooked by Dohm et al. (2008), probably because GC-rich areas are rare (especially in simpler organisms). Even in humans, it is hard to spot this effect if counts are binned by GC quantiles instead of GC values. Nevertheless, it is this departure from linearity that allowed pinpointing an optimal scale - the fragment size. In that, unimodality gives us important clues as to the causes of the GC bias.

While we have described other sequence-related biases, we believe they are not driving the strong coverage GC biases. These include an increased coverage when the ends are AT rich, and location-specific fragmentation biases near the fragment ends. We have shown that the end-effects, as measured on the 5’ end, are far weaker than the effect from the full fragment. They are also surprisingly negligible in the context of larger bins. Still, they might locally mitigate the fragment GC effect: the effect of fragment length on GC curve seems to be associated with these biases.

Our conclusions seem to complement those of Aird et al. (2011). If PCR is the major source of the GC bias, we would expect GC of the full fragment to be associated with the bias, rather than the GC of one or both reads. We have shown this is indeed the case. Moreover, data-sets generated according to a PCR-free protocol (Kozarewa et al. 2009) and an optimized PCR protocol (Aird et al. 2011) both display a reduced GC bias (Supplementary Figure S4, S5). It should be noted that even these optimized PCR protocols can still display significant biases and may require GC correction.

Our refined description of the GC effect is of practical value for GC correction. First of all, the non-linearity of the GC effect is a warning sign regarding two-sample correction methods. In the main example we study, the pair of normal and tumor samples do not have the same GC curves. We have seen this in additional data sets as well. Using normal counts to correct tumor counts could sometimes produce GC-related artifacts, which might lead to faulty segmentations. The GC effects of samples should be carefully studied before such corrections are made.

A single sample correction for GC requires a model, and we demonstrate the importance of choosing the best model. Overlapping windows smaller than the fragment fail to remove the bulk of the GC effect. Similarly, using read coverage rather than fragment count hurts the correction. Instead, measuring fragment rate for single bp positions, decouples the GC modeling from the down-stream analysis. Thus it removes the lower threshold on the scale of analysis, providing single bp estimates, which can be later smoothed by the researcher as needed (or binned into uneven bins if needed). An important benefit of DNA-seq over
previous technologies is that simply repeating the experiment can increase the resolution of the analysis. Our model assures this increased resolution does not hurt the GC correction.

Unlike other bias correction methods, such as BEADS (Cheung et al. 2011), we generate weights (predicted fragment rates) for the genomic location rather than for the observed reads. Mappable genomic positions are stratified according to the GC of a hypothetical fragment, and rates per GC stratum are estimated by counting the fragments at those same positions. Estimating predicted rates for both covered and uncovered locations can help detect deletions, and these predicted rates form a natural input for downstream analysis using heterogeneous Poisson models. Another important novelty is the use of TV scores to determine the representative fragment length of each data-set, one that best fits the distribution of fragment-lengths and properly discards the fragment-end biases. This procedure can be critical when length information is unavailable (i.e for single-ended reads). A more detailed comparison to BEADS is found in the Supplementary Figures S9, S10.

In this work, we estimated DNA abundance from non-tumor genomes, implicitly assuming that abundance of DNA along the genome is uniform. It is true that copy number variation may occur in non-tumor sequences; these jumps are rare however, and by random sampling we hope to average over any large CN changes. That the windows are small should reduce the dependence between GC and specific positions in the genome. From our experience, estimating GC curves using small windows turned out to be surprisingly robust to copy number changes on tumor data (as displayed above). To extend this method to other applications or protocols would require identifying regions in which the signal of interest is not expected to vary, and perhaps co-estimation of the abundance and the GC effect. That said, for CN purposes there is enough data to get stable estimates of the GC effect.

Our prediction accounts for a large portion of the variation, but residual variation is still present. Additional inhomogeneities in fragment rates include unexplained hot spots or zero-counts, as well as milder low and high frequency variation in the counts. The first two categories may be due to errors in the annotation of the genome, or amplification artifacts. The latter point to existence of additional factors that affect fragment rates, which is to be expected. We have discussed additional sequenced-related biases, including fragmentation and AT preference. The tools developed here, primarily the total variation scores, allow analysts to further investigate these effects as needed. Nevertheless, by and large, our model successfully describes the bulk of the low-frequency variability which confounds segmentation to copy number regions.

One effect that we have not deeply explored is the relation between sequencing error probability and the GC effect. In the Supplementary Data, we have shown evidence that the global GC of the fragment can effect the sequencing error probability. Especially for longer reads, changing the parameterization of the mapping processes can sometimes produce different mappability patterns related to the GC composition. There have been reports (Nakamura
et al. 2011) that specific sequences in reads are more prone for errors, for example a GGC sequence. A better model for reads that are harder to sequence would allow better estimation of the fragment GC effect in the GC rich regions, and improve the accuracy of the corrections. Jointly correcting by the GC of the read as well as the GC of the fragment may be a useful approximation for this effect.

Our analysis focused only on DNA-seq data from human subjects, but results from this work can be extended. GC content biases were seen in additional experimental protocols using high throughput sequencing. (See Supplementary Figures S7, S8, and (Cheung et al. 2011) for similar correction approaches in ChIP-seq data). Some of these protocols focus on highly localized signals on the genome, and could also benefit from strand specific and un-even bin normalization. Moreover, when length of the fragments is constrained (exon-sequencing, RNA-seq), a model taking both GC and fragment length into account may prove important. Fitting the model for each application is a challenge; still we believe that all these applications can benefit from our refined GC model.

2.5 Supplementary Data

Supplementary Data are available at NAR Online: Supplementary figures S1-S10, Supplementary methods and Supplementary reference(Li et al. 2008).
Chapter 3

The shuffle estimator for explainable variance in fMRI experiments

This is the first of two chapters describing the study of human and primate vision using prediction models. Humans can recognize and locate objects under variable viewing conditions. To understand the non-linear cognitive processing required, scientists record from the visual system as it responds to natural images or scenes - inputs the visual system has evolved to comprehend. They try to formulate how the image stimulus is represented by the neuronal response by fitting a predictive model. On the observed sample, the model is expected to remove the effects of noise on the outcome, predicting the accurate measurements. Moreover, only few natural scenes can be presented in any single experiment, whereas the model allows to generalize the representation to other, comparable, scenes.

In this chapter we deal with telling apart the different sources for prediction error: modeling error versus the presence of noise. In computational neuroscience, the level of noise is the most important reason for variation in prediction accuracy within a functional area. Noise reflects not only the difficulty of measuring the response, but also the presence of unobserved factors other than the stimulus that influence the response. Valid estimates for the signal and noise levels are therefore critical for evaluating a neuroscience data-set, both before and after prediction model are fit.\(^1\)

We focus on estimating the proportion of signal variance in the total variance of neural activity measurements in functional MRI (fMRI). In this data, strong noise correlations complicate the estimation problem. The correlations may confound the neural responses corresponding to the stimuli and, If not properly taken into account, inflate the explainable variance estimates and suggest false possible prediction accuracies.

\(^1\)In the previous chapter we assumed the coverage was distributed around its mean as a Poisson, an assumption that seems to hold in experiments.
3.1. INTRODUCTION

We propose a novel method to estimate the explainable variance: the shuffle estimator. This estimator is non-parametric, unbiased, and built upon the random effect model reflecting the randomization in the fMRI data collection process. Leveraging symmetries in the measurements, our estimator is obtained by appropriately permuting the measurement vector in such a way that the noise covariance structure is intact but the explainable variance is changed after the permutation. This difference is then used to estimate the explainable variance. We validate the properties of the proposed method in simulation experiments. For the image-fMRI data, we show that the shuffle estimates can explain the variation in prediction accuracy for voxels within the primary visual cortex (V1) better than alternative parametric methods. This chapter is adapted from a manuscript with Bin Yu.

3.1 Introduction

Neuroscientists study how human perception of the outside world is physically encoded in the brain. Although the brain’s processing unit, the neuron, performs simple manipulations of its inputs, hierarchies of interconnected neuron groups achieve complex perception tasks. By measuring neural activities at different locations in the hierarchy, scientists effectively sample different stages in the cognitive process.

Functional MRI (fMRI) is an indirect imaging technique, which allows researchers to sample a correlate of neural activities over a dense grid covering the brain. FMRI measures changes in the magnetic field caused by flow of oxygenated blood; these blood oxygen-level dependent (BOLD) signals are indicative of neuronal activities. Because it is non-invasive, fMRI can record neural activity from a human subject’s brain while the subject performs cognitive tasks that range from basic perception of images or sound to higher-level cognitive and motor actions. The vast data collected by these experiments allows neuroscientists to develop quantitative models, encoding models (Dayan et al. 2001), that relate the cognitive tasks with the activity patterns these tasks evoke in the brain. Encoding models are usually fit separately to each point of the spatial activity grid, a voxel, recorded by fMRI. Each fitted encoding model extracts features of the perceptual input and summarizes them into a value reflecting the evoked activity at the voxel.

Encoding models are important because they can be quantitatively evaluated based on how well they can predict on new data. Prediction accuracy of different models is thus a yard-stick to contrast competing models regarding the function of the neurons spanned by the voxel (Carandini et al. 2005). Furthermore, the relation between the spatial organization of neurons along the cortex and the function of these neurons can be recovered by feeding the model with artificial stimuli. Finally, predictions for multiple voxels taken together create a predicted fingerprint of the input; these fingerprints have been successfully used for extracting information from the brain (so called “mind-reading” ,Nishimoto et al. 2011), and building
brain machine interfaces (Shoham et al. 2005). The search for simpler but more predictive encoding models is ongoing, as researchers try to encode more complex stimuli and predict higher levels of cognitive processing.

Because brain responses are not deterministic, encoding models cannot be perfect. A substantial portion of the fMRI measurements is noise that does not reflect the input. The noise may be caused by background brain activity, by non-cognitive factors related to blood circulation, or by the measurement apparatus. Regardless of the source, the noise cannot be predicted by encoding models that are deterministic functions of the inputs (Roddey et al. 2000). To reduce the effect of noise, the same input can be displayed multiple times within the input sequence and all responses to the same input averaged, in an experimental design called event-related fMRI (Josephs et al. 1997). See Pasupathy and Connor (1999a); Haefner and Cumming (2008) for examples, and Huettel (2012) for a review. Typically, even after averaging, the noise level is high enough to be a considerable source of prediction error. Hence it is standard practice to measure and report an indicator of the signal strength together with prediction success. We will focus on one such indicator, the proportion of signal variance in the total variance of the measurements. We call this quantity the explainable variance\(^2\), because it measures the proportion of variance that can be explained by a deterministic model. The comparison of explainable variance with prediction success (Roddey et al. 2000; Sahani and Linden 2003) informs how much room is left on this data for improving prediction through better models. Explainable variance is also an important quality control metric before fitting encoding models, and can help choose regularization parameters for model training.

In this work we develop a new method to estimate the explainable variance in fMRI responses, and use it to reanalyze data from an experiment conducted by the Gallant lab at UC Berkeley (Kay et al. 2008; Naselaris et al. 2009). Their work examines the representation of visual inputs in the human brain using fMRI by ambitiously modeling a rich class of images from natural scenes rather than artificial stimulus. An encoding model was fit to each of more than 10,000 voxels within the visual cortex. The prediction accuracy of their fitted models on a separate validation image set were surprisingly high given the richness of the input class, inspiring many studies of rich stimuli class encoding (Pasley et al. 2012; Pereira et al. 2011). Still, accuracy for the voxels varied widely (see Figure 3.2), and more than a third of the voxels had prediction accuracy not significantly better than random guessing. Researchers would like to know whether accuracy rates reflect (a) overlooked features which might have improved the modeling, or instead reflect (b) the noise that cannot be predicted regardless of the model used. As we show in this chapter, valid measures of explainable variance can shed light on this question.

\(^2\)This proportion is known by other names depending on context, such as intraclass correlation, effect-size, and pseudo \(R^2\).
Measuring explainable variance on correlated noise

We face the statistical problem of estimating the explainable variance, assuming the measurement vector is composed of a random mean-effects signal evoked by the images with additive auto-correlated noise (Scheffé 1959). In fMRI data, many of the sources of noise would likely affect more than one measurement. Furthermore, low frequency correlation in the noise has been shown to be persistent in fMRI data (Fox and Raichle 2007). Ignoring the correlation would greatly bias the signal variance estimation (see Figure 3.7 below), and would cause us to over-estimate the explainable variance. This over-estimation of signal variance may be a contributing factor to replicability concerns raised in neuroscience (Vul et al. 2009).

Classical analysis-of-variance methods account for correlated noise by (a) estimating the full noise covariance, and (b) deriving the variances of the signal and the averaged noise based on that covariance. The two steps can be performed separately by methods of moments (Scheffé 1959), or simultaneously using restricted maximum likelihood (Laird et al. 1987). In both cases, some parametric model for the correlation is needed for the methods to be feasible, for example a fast decay is assumed in Woolrich et al. (2001). These approaches, however, are sensitive to misspecifications of the correlation parameters. In fMRI signals, the correlation of the noise might vary with the specifics of the preprocessing method in a way that is not easy to parametrize. As we show in Section 3.6, if the autocorrelation model is too simplistic it might not capture the correlation well and over-estimate the signal; on the other hand, if it is too flexible the noise might be over-estimated, and furthermore the numeric optimizations involved in estimating the correlation might sometimes fail to converge.

An alternative way (Sahani and Linden 2003; Hsu et al. 2004) to handle the noise correlation when estimating variances is to restrict the analysis to measurements that, based on the data collection, should be independent. Many neuroscience experiments are divided into several sessions, or blocks, to better reflect the inherent variability and to allow the subject rest. Fewer have a block design, where the same stimulus sequence is repeated for multiple blocks. Under block design the signal level can be estimated by comparing repeated measures across different blocks: regardless of the within-block-correlation, the noise should decay as $1/b$ when averaged over $b$ blocks with the same stimulus sequence. Block designs, however, are quite limiting for fMRI experiments, because the long reaction time of fMRI limits the number of stimuli that can be displayed within an experimental block (Huettel 2012). The methods above also do not use repeats within a block to improve their estimates. These problems call for a method that can make use of patterns in the data collection to estimate the signal and noise variances under less restrictive designs.

We introduce novel variance estimators for the signal and noise levels, which we call shuffle estimators. Shuffle estimators resemble bias correction methods: we think of the noise component as a "bias" and try to remove it by resampling. The key idea is to artificially create a second data vector that will have similar noise patterns as our original data. We do
3.2 Preliminaries

3.2.1 An FMRI Experiment

In this section we describe an experiment carried out by the Gallant lab at UC Berkeley (Kay et al. 2008), in which a human subject viewed natural images while scanned by fMRI. The two primary goals of the experiment were (a) to find encoding models that have high predictive accuracy across many voxels in the early visual areas; and (b) to use such models to identify the input image, from a set of candidate images, based on the evoked brain patterns.

3 We use data from subject S1 in Kay et al.
The experiment created the first non-invasive machinery to successfully identify natural images based on brain patterns, and its success spurred many more attempts to encode and decode neural activities evoked by various cognitive tasks (Pasley et al. 2012; Pereira et al. 2011). We focus only on the prediction task, but note that gains in prediction would improve the accuracy of identification as well. A complete description of the experiment can be found in the supplementary materials of the original paper (Kay et al. 2008). This is background for our work, which begins in Section 3.2.2.

The data of this experiment are composed of the set of natural images, and the fMRI scans recorded corresponding to the images. The images were sampled from a library of gray-scale photos depicting natural scenes, objects, etc. Two non-overlapping random samples were taken: the training sample (1750 images) was used for fitting the models; and the validation sample (\(m = 120\) images) was used for measuring the prediction accuracy. Images were sequentially displayed in a randomized order, each image appearing multiple times (\(n = 13\)). Blood oxygen-level dependent contrasts, which measure a correlate of neural activity, were continuously recorded by the fMRI across as the subject watched the images from a three dimensional grid covering the visual cortex. For each voxel in the grid, the responses were temporally discretized so that a single value (per voxel) was associated with a single displayed image.

Data from the training sample was used to fit a quantitative receptive field model for each voxel, describing the fMRI response as a function of the input image. Here is a brief overview; more details on these V1 encoding models can be found in Vu et al. (2011). The model was based on multiple Gabor filters capturing spatial location, orientation, and spatial-frequency of edges in the images (see Figure 3.1). Because of the tuning properties of the Gabor energy filters, this filter set is typically used for representing receptive fields of mammalian V1 neurons. Gabor filters (\(d = 10409\) filters) transformed each image into a feature vector in \(\mathbb{R}^d\). For each of \(Q\) voxels of interest, a linear weight vector relating the features to the measurements was estimated from the training data. Together, the transformation and linear weight vector result in a prediction rule that maps novel images to a real-valued response per voxel.

In their paper, Kay et al. measured prediction accuracy by comparing observations from the validation sample with the predicted responses for those images. The validation data consisted of a total of \(T = 1560\) measurements (per voxel): \(m = 120\) different images, each repeated \(n = 13\) times. Let \(Y_t^{(r)} \in \mathbb{R}\) denote the measured fMRI activity at voxel \(r\) at time \(t\). Repeated measurements of the same image were averaged to reduce noise:

\[
\bar{Y}_j^{(r)} = \text{avg}_{t:h(t) = j} Y_t^{(r)}, \quad j = 1, ..., m,
\]

where \(h(t) \in \{1, ..., m\}\) indexes the image that was shown at time \(t\).

Let \(f_1^{(r)}, ..., f_m^{(r)}\) be the sequence of predictions for voxel \(r\), and \(\bar{f}^{(r)}\) their average. A single
3.2. PRELIMINARIES

Figure 3.1: A Gabor-based encoding model for natural images. A cartoon depicting the encoding models that were used for the fMRI experiment. Natural images (a) were transformed into a vector of 10409 features; features (b) measure the combined energy from two linear Gabor filters -tuned for specific spatial frequency, location in the image, and orientation - with complementary phases. These features are combined according to a linear weight vector (c) that was fit separately for each voxel. The predicted response of a voxel for an image is the weighted sum of the features representing the image (d). The linear weights were fit on the training data consisting of responses to 1750 images. This cartoon was adapted from Kay et al. (2007).

value per voxel summarizes prediction accuracy. That is,

$$\text{Corr}^2[(f_j^{(r)})_{j \leq m}; (\hat{Y}_j^{(r)})_{j \leq m}] := \frac{\left(\sum_{j=1}^{m} (f_j^{(r)} - \bar{f}^{(r)}) (\hat{Y}_j^{(r)} - \bar{Y}^{(r)})\right)^2}{\sum_{j=1}^{m} (f_j^{(r)} - \bar{f}^{(r)})^2 \sum_{j=1}^{m} (\hat{Y}_j^{(r)} - \bar{Y}^{(r)})^2}.$$  

In Figure 3.2 we show examples of voxels with low, intermediate, and high prediction accuracies, and a histogram of the accuracy for all 1250 voxels located within the V1 area. We can drop the superscript $r$ because each voxel is analyzed separately.

3.2.2 Correlation in the data

The goal of our work is to separate the two factors that determine the accuracy of prediction rules: the adequacy of the feature set or the linear model; and the noise level. Explainable variance represents the optimal accuracy if prediction was unrestricted by the choice of features and model.
Figure 3.2: **Prediction accuracy for V1 voxels.** Predicted vs. observed average responses for three voxels in the V1 area, reflecting poor (a), medium (b), and high (c) prediction accuracies. Each point depicts the predicted response (x-axis) and the observed average response (across all repeats) for an image of the validation sample (m=120 images). (d) Histogram of prediction accuracy for all 1250 V1 voxels.

We validate our explainable variance estimators by showing the estimators account well for the differences in prediction accuracy between Q = 1250 voxels within the primary visual cortex (V1). This analysis depends on the assumption that the observed variation between voxels is primarily due to differences in the level of the signal-to-noise in the validation data, rather than, for example, differences in the adequacy of the feature set underlying the prediction models. Once the estimators are validated on this controlled setting, explainable variance can be used more broadly, for example to compare the predictability levels of different functional areas.

Since we intend to use the validation data set to estimate the explainable variance, we now give a few more details on how it was collected. Recall that the validation data consisted of m = 120 images each repeated several times (see Figure 3.3 a). This data was recorded in 10 separate sessions, so that the subject could rest between sessions; the fMRI was re-calibrated at the beginning of each session. Each session contained all presentations of 12 different images. A pseudo-random integer sequence ordered the repeats within a session\(^4\).

\(^4\)The pseudo-random sequence allocated spots for 13 different images; no image was shown in the last category and the responses were discarded.
When we measure correlation across many voxels, it appears that the design of the experiment induces strong correlation in the data. To see this, in Figure 3.3 (b-c) we plot the correlation between measurements at different time slots (each time slot is represented by the vector of Q=1250 measurements). This gives us a gross representation of the correlation for individual voxels, including both noise driven and possibly stimuli-driven correlations. Clearly there are strong correlations between time-slots within a block, but no observable correlations between blocks. As these within-block correlation patterns do not correspond to the stimuli schedule that is randomized within a block, we conclude the correlations are largely due to noise. These noise correlations need to be taken into account to correctly estimate the explainable variance.

3.2.3 A probability model for the measurements.

We introduce a probabilistic model for the measurements $Y := (Y_t)_{t=1}^T$ at a single voxel. $Y$ is modeled as a random effects model with additive, correlated noise (Williams 1952). We assume the observed response is the sum of two independent random processes: the signal process, which is random due to sampling of images into the validation set, and the noise process describing fluctuations unrelated to the stimuli. Additivity of noise is considered a good approximation for fMRI event related designs and is commonly used (Buracas and Boynton 2002). The random effects model accounts for the generalization of prediction accuracy from the validation sample to the larger population of natural images.

Signal

Images are shown in a long sequence, in which each of the $m$ images in the random sample is repeated multiple times. The order of presentation is described by the design matrix $X \in \{0, 1\}^{T \times m}$ (see Figure 3.3). Each row vector $X_t$ has a single 1 entry, which identifies the image shown at time $t$,

$$X_{t,j} = 1$$ if the j’th image in the sample is displayed at time $t$.

For now, the sampling of images is conveyed through their effects on the measurement. We use a homogenous set of random variables (Owen 2007) to represent the mean responses to the images in the sample. Let $A_j$ be the mean of the observed responses to the j’th image in the sample. By taking

$$A_j \sim (0, \sigma_A^2)$$ for $j = 1, ..., m$,

we mean that $A_j$’s are iid with mean 0 and variance $\sigma_A^2 \geq 0$, but are not necessarily normally distributed. In the experiment we are analyzing, the randomness originates from sampling
3.2. PRELIMINARIES

Figure 3.3: Data acquisition for the validation data set. The responses in the validation set were collected in 10 separate sessions (blocks). Both the design matrix for the experiment and the noise correlation are influenced by this structure. In (a), the transposed design matrix $X'$ is shown. This matrix records which image (y-axis) is displayed at each time slot $t$ (x-axis). Separate sets of 12 images were repeated $n = 13$ times within each block, whereas no image was recorded in more than one block. In (b-d) temporal autocorrelation is displayed, as measured between a single time-point ($t^* = 40, 140, 240$ for b,c,d respectively), and all others points. The $t = t^*$ point is marked by blue vertical lines. On average, strong but non-smooth correlation are found within the blocks, but separate blocks seem uncorrelated. Note that we depict the aggregate correlation of all voxels here, but cannot from this infer the noise correlation of any specific voxel. Furthermore, correlation depicted here is due to both noise and signal.

A large (infinite) population of images. We use $\mathbf{A} := (A_j)_{j=1}^m$ for the random-effect column vector, $\bar{A} := \frac{1}{m} \sum_{j=1}^m A_j$ for the sample mean, and $s_A^2 := \frac{1}{(m-1)} \sum_{j=1}^m (A_j - \bar{A})^2$ for the sample variance of the random effects.

$X\mathbf{A}$ is the $T$-dimensional random signal vector, denoting the random effect at each measurement. To index the effect corresponding to time $t$, we use the shorthand $A(t) := X_t \mathbf{A}$. 
3.2. PRELIMINARIES

Noise

We assume the measurement noise process \( \epsilon = (\epsilon_t)_{t=1}^T \) is independent of the random signal vector \( XA \). We further assume the noise elements \( \epsilon_t \) have 0 mean, \( \sigma_\epsilon^2 \geq 0 \) variance, but may be auto-correlated. The unknown correlation, denoted by a matrix \( \Sigma \in \mathbb{R}^{T \times T} (\text{diag}(\Sigma) = 1) \), captures the slow-changing hemodynamics of the BOLD and the effects of preprocessing on the BOLD signals. Hence,

\[
E_\epsilon[\epsilon_t] = 0; \quad \text{cov}(\epsilon_t, \epsilon_u) = \sigma_\epsilon^2 \Sigma_{tu} \quad \Sigma_{tt} = 1,
\]

or in matrix notation \( \text{cov}[\epsilon] = \sigma_\epsilon^2 \Sigma \).

Model for observed responses

We are now ready to introduce the observed response (column) vector \( Y \in \mathbb{R}^T \) as follows:

\[
Y = XA + \epsilon \quad \text{(3.2)}
\]

and for a single time slot \( t \)

\[
Y_t = A(t) + \epsilon_t.
\]

Response covariance

The model involves two independent sources of randomness: the image sampling, modeled by the random effects \( A \), and the measurement noise \( \epsilon \). Assuming independence between \( A \) and \( \epsilon \), the covariance of \( Y_t \) and \( Y_u \) amounts to adding the individual covariances

\[
\text{cov}_{A,\epsilon}(Y_t, Y_u) = \text{cov}_A(A(t), A(u)) + \text{cov}_\epsilon(\epsilon_t, \epsilon_u) = \sigma_A^2 1(Y_t=x_u) + \sigma_\epsilon^2 \Sigma_{tu}. \quad \text{(3.3)}
\]

The first term on the RHS shows that treatment (random) effects are uncorrelated if they are based on different inputs, but are identical if based on the same input, with a variance of \( \sigma_A^2 \). In matrix form, we get:

\[
E_{A,\epsilon}[Y] = 0; \quad \text{cov}_{A,\epsilon}(Y) = \sigma_A^2 XX' + \sigma_\epsilon^2 \Sigma. \quad \text{(3.4)}
\]

3.2.4 Explainable variance and variance components

We are ready to define explainable variance, a scaled version of the treatment variance \( \sigma_A^2 \). Explainable variance is relevant to the performance of prediction models, a property we will discuss in Section 3.4.
Recall that $\bar{Y}_j$ are the averaged responses per image ($j = 1, \ldots, m$ for the images in our sample), and let $\bar{Y} = \frac{1}{T} \sum_{t=1}^{T} Y_t$ be the global average response. Then the sample variance of averages is

$$MS_{bet} := \frac{1}{m-1} \sum_{j=1}^{m} (\bar{Y}_j - \bar{Y})^2.$$  \hfill (3.5)

The notation $MS_{bet}$ refers to the mean-of-squares between treatments. Let us define the total variance $\sigma^2_Y$ as the population mean of $MS_{bet}$,

$$\bar{\sigma}^2_Y := \mathbb{E}_{A,\epsilon}[MS_{bet}].$$  \hfill (3.6)

Note that $\bar{\sigma}^2_Y$ is not strictly the variance of any particular $\bar{Y}_j$; indeed, the variance of $\bar{Y}_j$ is not necessarily equal for different $j$'s. Nevertheless, we will loosely use the term variance here and later, owing to the parallels between these quantities and the variances in the iid noise case.

The average across repeats $\bar{Y}_j$ is composed of a signal part ($A_j$) and an average noise part ($\bar{\epsilon}_j$); similarly $\bar{Y}$ is composed of $\bar{A}$ and $\bar{\epsilon}$. By partitioning the $MS_{bet}$, and taking expectations over the sampling and the noise, we get

$$\mathbb{E}_{A,\epsilon}[MS_{bet}] = \mathbb{E}_{A}[\frac{1}{m-1} \sum_{j=1}^{m} (A_j - \bar{A})^2] + \mathbb{E}_{\epsilon}[\frac{1}{m-1} \sum_{j=1}^{m} (\bar{\epsilon}_j - \bar{\epsilon})^2],$$  \hfill (3.7)

where the cross-terms cancel because of the independence of the noise from the sampling. We can call the expectation of the second term the noise level, or $\bar{\sigma}_{\epsilon}^2$, and get the following decomposition

$$\bar{\sigma}^2_Y = \sigma^2_A + \bar{\sigma}_{\epsilon}^2.$$  \hfill (3.8)

In other words, the signal variance $\sigma^2_A$ and the noise level $\bar{\sigma}_{\epsilon}^2$ are the signal and noise components of the total variance.

Finally, we define the proportion of explainable variance to be the ratio

$$\omega^2 := \sigma^2_A / \bar{\sigma}^2_Y.$$  

Explainable variance measures the proportion of variance due to signal in the averaged responses; estimating it is the goal of this work. Note that by definition, $\omega^2$ is restricted to $[0, 1]$.

In order to estimate $\omega^2$, we need estimators for $\sigma^2_Y$ and $\sigma^2_A$. Whereas $\sigma^2_Y$ can be directly estimated from the sample, to estimate $\sigma^2_A$ we need a method to separate the signal from the noise. In the following, we propose a method to tell the two apart using their different covariance structures. First, though, we will develop some technical algebraic identities important for the estimation procedure. Some readers might prefer to skip directly to Section 3.3.

\footnote{In practice, this is true for the individual measurements $Y_t$ as well. We chose $\Sigma_{tt} = 1$ for illustration reasons.}
3.2. PRELIMINARIES

3.2.5 Quadratic contrasts

In this subsection we express $MS_{bet}$ as a quadratic contrast of the full data vector $Y$. This contrast highlights the relation between $\bar{\sigma}_Y^2$ or $\bar{\sigma}_e^2$ with both the design $XX'$ and the measurement correlations $\Sigma$, and produces algebraic descriptions to be used for deriving the Shuflfe estimator. These are simple extensions of classical treatment of variance components (Townsend and Searle 1971).

Denote $B := XX'/n$, the $\mathbb{R}^{T \times T}$ matrix in which

$$B_{tu} = \begin{cases} \frac{1}{n} & \text{if } X_t = X_u, \\ 0 & \text{otherwise}. \end{cases} \quad (3.9)$$

$B$ is an averaging matrix, meaning that multiplication of a measurement vector by $B$ replaces each element in the vector by the treatment average, as in

$$(BY)_t = \bar{Y}_{h(t)}. \quad (3.10)$$

It is easy to check that $B = B'$ and $B = B^2$. Also let $G \in \mathbb{R}^{T \times T}$, $G_{tu} = 1/T$ for $t, u = 1, ..., T$, be the global average matrix, so that $(GY)_t = \bar{Y}$, $t = 1, ..., T$. We can now express $MS_{bet}$ as a quadratic expression of $Y$

$$MS_{bet} = \frac{1}{(m-1)n} \| (B - G)Y \|^2, \quad (3.11)$$

or more generally as a quadratic function of any input vector

$$MS_{bet}(\cdot) := \frac{1}{(m-1)n} \| (B - G)(\cdot) \|^2. \quad (3.12)$$

By replacing the $MS_{bet}$ with its quadratic form, a relation is exposed between the total variance, the design, and the correlation of the noise:

$$\bar{\sigma}_Y^2 = \mathbb{E}_{A, e}[MS_{bet}(Y)] = \frac{1}{(m-1)n} \mathbb{E}_{A, e}[tr ((B - G)(Y'Y)(B - G))]$$

$$= \frac{1}{(m-1)n} tr ((B - G)\text{cov}_{A, e}(Y)).$$

The signal effect and noise are additive, hence

$$\frac{1}{(m-1)n} tr ((B - G)\text{cov}(Y)) + \frac{1}{(m-1)n} tr ((B - G)\text{cov}_{e}(Y)).$$

Substituting $\text{cov}(Y) = n\sigma_A^2 B$ and $\text{cov}_{e}(Y) = \sigma_e^2 \Sigma$,

$$\frac{1}{(m-1)n} tr ((B - G)(n\sigma_A^2 B)) + \frac{1}{(m-1)n} tr ((B - G)\sigma_e^2 \Sigma).$$

$$= \frac{1}{(m-1)n} \sigma_A^2 tr (B - G) + \frac{1}{(m-1)n} \sigma_e^2 tr ((B - G)\Sigma)$$

$$= \sigma_A^2 + \frac{1}{(m-1)n} \sigma_e^2 tr ((B - G)\Sigma).$$
3.3. SHUFFLE ESTIMATORS FOR SIGNAL AND NOISE VARIANCES

**Derivation 1.** *Under the model described in Section 3.2.3,*

\[
\sigma_Y^2 = \sigma_A^2 + \frac{1}{(m-1)n} \sigma^2 \text{tr} ((B - G)\Sigma). \tag{3.13}
\]

As a direct consequence of (3.8, 3.13) we get an exact expression for the noise level

\[
\bar{\sigma}_\epsilon^2 = \frac{1}{(m-1)n} \sigma^2 \text{tr} ((B - G)\Sigma). \tag{3.14}
\]

This expression clarifies how \( \bar{\sigma}_\epsilon^2 \) depends on the design, the noise variance and the noise auto correlation. As expected, \( \bar{\sigma}_\epsilon^2 \) scales linearly with the noise variance of the individual measurements \( \sigma^2 \). More interesting is that \( \bar{\sigma}_\epsilon^2 \) depends linearly on \( \text{tr} ((B - G)\Sigma) \), or the interplay between the design and the noise auto-correlation.

Note that if the within treatment noise is uncorrelated, this expression simplifies to a classical ANOVA result. Uncorrelated noise within treatments manifests, in a properly sorted version of \( \Sigma \), as small \( n \times n \) identity blocks. Therefore \( \text{tr} ((B - G)\Sigma) = (m - 1)\sigma^2 \) and \( \bar{\sigma}_\epsilon^2 = \sigma^2 / n \). In that case \( \sigma_Y^2 = \sigma_A^2 + \sigma^2 / n \), and by plugging in an estimator of \( \sigma^2 \), we can directly estimate \( \bar{\sigma}_\epsilon^2 \) and \( \sigma_A^2 \). The estimator for \( \omega^2 \) is

\[
\hat{\omega}^2 = 1 - \frac{1}{F},
\]

with \( F \) being the standard F statistic. This is the method-of-moments estimator described fully in Section 3.6.1.

On the other hand, when some correlations within repeats are greater than 0, \( \sigma^2 / n \) underestimates the noise level and inflates the explainable variance. In the next section we introduce the shuffle estimators which can deal with correlated noise.

### 3.3 Shuffle estimators for signal and noise variances

In this section we propose new estimators called the shuffle estimators for the signal and noise levels, and for the explainable variance. As in (3.8), \( \sigma_Y^2 = \sigma_A^2 + \bar{\sigma}_\epsilon^2 \), but the noise level \( \bar{\sigma}_\epsilon^2 \) is a function of the (unknown) measurement correlation matrix \( \Sigma \). Using shuffle estimators we can estimate \( \sigma_A^2 \) and \( \bar{\sigma}_\epsilon^2 \) without having to estimate the full \( \Sigma \) or imposing unrealistically strong conditions on it.

The key idea is to artificially create a second data vector that will have similar noise patterns as our original data (see Figure 3.4). We do this by permuting, or *shuffling*, the original data with accordance to symmetries that are based on the data collection. In Section 3.3.1 we formalize the definition of such permutations that conserve the noise correlation, and give
plausible examples for neuroscience measurements. In Section 3.3.2 we compare the variance of averages ($MS_{bet}$) of the original data (Figure 3.4 b), with the same contrast computed on the shuffled data (c). Because repeated measures for the same image are shuffled into different categories, the variance due to signal will be reduced in the shuffled data. We derive an unbiased estimator for signal variance $\sigma_A^2$ based on this reduction in variance, and use the plug-in estimators to estimate $\bar{\sigma}_A^2$ and $\omega^2$.

![Diagram](image)

Figure 3.4: Cartoon of the shuffle estimator. (a) Data is generated according to a predetermined design, with each color representing repeats of a different image. (b) Repeats of each image are averaged together and the sample variance is computed on these averages. (c) Data is shuffled by $P$, in this example reversing the order. Now measurements which do not originate from the same repeat are averaged together ($\bar{Y}_j$'s), and the sample variance of the new averages is computed. These averages should have a lower variance in expectation, and we can calculate the reduction amount $\alpha = \frac{1}{m-1} tr ((B - G)PBP^T)$, where $B = XX'/n$. (d) The shuffle estimator for signal variance is the difference between the two sample variances, after correction of $1 - \alpha$.

### 3.3.1 Noise conserving permutation for Y with respect to design X

A prerequisite for the shuffle estimator is to find a permutation that will conserve the noise contribution to $\sigma_Y^2$. We will call such permutations noise-conserving w.r.t to $X$. 
Recall (3.14),

\[ \tilde{\sigma}_e^2 = \frac{1}{(m-1)n} \text{tr}((B - G) (\sigma_e^2 \Sigma)), \]

where \( \sigma_e^2 \Sigma = \text{cov}_i [Y] \) as before. Let \( P \in \mathbb{R}^{T \times T} \) be a permutation matrix. Then

**Definition 1.** \( P \) is noise conserving w.r.t \( X \), if

\[ \text{tr} ((B - G) P \sigma_e^2 \Sigma P') = \text{tr} ((B - G) \sigma_e^2 \Sigma). \]  

(3.15)

Equivalently,

\[ \text{tr} ((B - G) \text{cov}_i [P \cdot Y]) = \text{tr} ((B - G) \text{cov}_i [Y]). \]

Although we define the noise conserving property based on the covariance, replacing the covariance with the correlation matrix \( \Sigma \) would not change the class of noise conserving permutations.

Let us take a look at important cases of noise-conservation. Though most useful noise-conserving permutations will be derived from assumptions regarding the noise correlation \( \Sigma \), the noise-conservation property can be derived from the design \( X \) as well. This can be seen in the first example.

**Trivial noise-conserving permutations**

A permutation \( P \) that simply relabels the treatments is not a desirable permutation, even though it is noise-conserving. We call such permutations trivial:

**Definition 2.** A permutation \( P \), associated with permutation function \( g_P : \{1...T\} \to \{1...T\} \), is trivial if

\[ X_t = X_u \Rightarrow X_{g_P(t)} = X_{g_P(u)}, \quad \forall t, u. \]  

(3.16)

It is easy to show that for trivial \( P \), \( MS_{\text{bet}}(PY) = MS_{\text{bet}}(Y) \).

**Noise conserving permutations based on symmetries of \( \Sigma \)**

A useful class of non-trivial noise conserving permutations is the class of symmetries in the correlation matrix \( \Sigma \): a symmetry of \( \Sigma \) is a permutation \( P \) such that \( PS\Sigma P' = \Sigma \). If \( P \) is a symmetry of \( \Sigma \), then \( P \) is noise-conserving regardless of the design. Here are three important general classes of symmetries which are commonly applicable in neuroscience.

1. **Uncorrelated noise.** The obvious example is the uncorrelated noise case \( \Sigma = I \) where all responses are exchangeable. Hence any permutation is noise-conserving.
2. **Stationary time series** Neuroscience data is typically recorded in a long sequence containing a large number of serial recordings at constant rates. It is natural to assume that correlations between measurements will depend on the time passed between the two measures, rather than on the location of the pair within the sequence. We call this the *stationary time series*. Under this model $\Sigma$ is a Toeplitz matrix parameterized by $\{\rho_d\}_{d=0}^{T-1}$, the set of correlation values $\Sigma_{t,u} = \rho_{|t-u|}$. Though the correlation values $\rho_d$’s are related, this parameterization does not enforce any structure on them. This robustness is important in the fMRI data we analyze. For this model, a permutation that *reverses* the measurement vector is noise conserving

$$(PY)_t = Y_{T+1-t}.$$  

This is the permutation we use on our data in Section 3.6.

The shift operators of the form $(PY)_t = (Y)_{t+k}$ define a transformation that, up to edge effects, can be considered noise conserving.

3. **Block effect models** Another important case is when measurements are collected in distinct sessions, or blocks. Measurements from different blocks are assumed independent, but measurements within the same block may be correlated, perhaps because of calibration of the measurement equipment. We index the block assignment of time $t$ with $\beta(t)$. A simple parameterization for noise correlation would to let $\Sigma_{t,u} = \zeta(\beta(t), \beta(u))$ depend only on the block identity of measurements $t$ and $u$. We call this the *block structure*. Under the block structure, any permutation $P$ (associated with function $g_P$) that maintains the identity of blocks, meaning

$$\beta(t) = \beta(u) \Rightarrow \beta(g_P(t)) = \beta(g_P(u)) \quad (3.17)$$

would be noise-conserving w.r.t. any $X$.

The scientist is given much freedom in choosing the permutation $P$, and should consider both the variance of the estimator and the estimator’s robustness against plausible noise-correlation structures. Establishing criteria for choosing the permutation $P$ is the topic of current research.

### 3.3.2 Shuffle estimators

We can now state the main results. From the following lemma we observe that every noise-conserving permutation establishes a mean-equation with two parameters: $\sigma_A^2$ and $\bar{\sigma}_\varepsilon^2$. In the following lemma, we show that the coefficient for $\sigma_A^2$ is a constant

$$\alpha = \alpha(X,P) = \frac{1}{m-1} tr \left((B-G)(PB\varepsilon')\right),$$
which depends only on the design $B = XX'/n$ and the permutation $P$ - both known to the scientist. The size of $\alpha$ reflects how well $P$ "mixes" the treatments; the greater the mix, the smaller $\alpha$.

**Lemma 1.** If $P$ is a noise-conserving permutation for $Y$, then

1. $\mathbb{E}_{A,\epsilon}[MS_{\text{bet}}(PY)] = \alpha \sigma^2_A + \bar{\sigma}^2_{\epsilon}$.

2. $\alpha \leq 1$, and the inequality is strict iff $P$ is non-trivial.

**Proof.**

1. Using similar algebra as in 3.13, the expectation $\mathbb{E}_{A,\epsilon}[MS_{\text{bet}}(PY)]$ can be partitioned into a term depending on the sampling covariance $cov_A(PY)$ and a term depending on the noise covariance $cov_{\epsilon}(PY)$. Since $P$ is noise-conserving, for the noise term:

$$cov_{\epsilon}(PY) = \bar{\sigma}^2_{\epsilon}.$$ 

As for the sampling:

$$cov_A(PY) = Pcov_A(Y)P' = \sigma^2_A P(nB)P'.$$

Hence,

$$\frac{1}{(m-1)n} \sigma^2_A tr \left( (B - G)(P(nB)P') \right) = \alpha \sigma^2_A.$$

2. For the unpermuted vector, the sampling component $cov_A(Y)$ is $\sigma^2_A$ (3.13). Hence for $P = I$ the $T \times T$ identity matrix, we have $\alpha(P, X) = 1$. For all other $P$'s, note that the global mean term ($G$) is unaffected by the permutation ($PG = G$) or the averaging ($BG = G$), so it remains unchanged.

From the Cauchy-Schwarz inequality,

$$tr \left( B(PBP') \right) \leq tr(BB) = tr(B)$$

with equality achieved iff $PBP' = B$ because $P$ is unitary and $B$ a projection. Recall that $P$ is trivial if $P$ reorders measurements within categories and renames categories. It is easy to check that $PBP' = B$ iff $P$ is trivial. For trivial $P$'s, we again get equations similar to 3.13, so $\alpha = 1$.

For any **non-trivial** permutation $B \neq PBP'$, in which case the CS-inequality is strict resulting in $\alpha < 1$. 

The consequence of the second part of the lemma is that for any non-trivial $P$, we get a second mean-equation, which is linearly independent from the equation based on the original
3.3. SHUFFLE ESTIMATORS FOR SIGNAL AND NOISE VARIANCES

data (because $\alpha < 1$). In other words, for a non-trivial $P$, the equation set
\[
\begin{align*}
\mathbb{E}_{A,X}[MS_{\text{bet}}(Y)] &= \sigma^2_A + \overline{\sigma}^2_e \\
\mathbb{E}_{A,X}[MS_{\text{bet}}(PY)] &= \alpha \sigma^2_A + \overline{\sigma}^2_e
\end{align*}
\] (3.18)
has a unique solution. Solving the two equations above, we arrive at an unbiased estimator for the signal variance.

**Definition 3.** The shuffle estimator for the signal variance is defined as
\[
\hat{\sigma}^2_A = \frac{MS_{\text{bet}}(Y) - MS_{\text{bet}}(PY)}{1 - \alpha}.
\] (3.19)

Finally, we can plug in $\hat{\sigma}^2_A$ and $\hat{\sigma}^2_Y = MS_{\text{bet}}(Y)$ to get the shuffle estimator for the explainable variance $\omega^2$
\[
\hat{\omega}^2 = \frac{\hat{\sigma}^2_A}{MS_{\text{bet}}(Y)} = \frac{1}{1 - \alpha} \left( 1 - \frac{MS_{\text{bet}}(PY)}{MS_{\text{bet}}(Y)} \right).
\]

Remarks

1. An unbiased estimator of the noise level $\hat{\sigma}^2_e$ can be derived from Equations 3.19 and 3.8
\[
\hat{\sigma}^2_e = MS_{\text{bet}} - \hat{\sigma}^2_A.
\]

2. Because $\hat{\sigma}^2_A$ estimates a non-negative quantity, it is preferable to restrict $\hat{\sigma}^2_A$ to non-negative values by taking $\hat{\sigma}^2_A = \max\{0, \hat{\sigma}^2_A\}$.

3. The estimator is consistent under proper decay of the dependence. This statement is conditional on the asymptotic setup: explainable variance typically changes as the number of measurements ($T$) increases. Nevertheless, the shuffle estimator is consistent for a sequence of data sets (indexed by $k = 1, 2, \ldots$) of growing sizes ($T(k) \to \infty$) for which total variance and explainable variance converge if (a) the number of treatments $m(k)$ grows to $\infty$ and (b) the dependence decays. For $Y$ distributed as a multivariate gaussian\(^6\), a sufficient condition for (b) can be given in terms of eigenvalues of $\Sigma$:

(b*) The largest $m - 1$ eigenvalues $\lambda_{(1)}(k), \ldots, \lambda_{(m-1)}(k)$ of the noise correlation matrix $\Sigma(k)$ satisfy
\[
\frac{1}{n^2(m-1)^2} \sum_{i=1}^{m-1} \lambda_{(i)}^2(k) \to 0 \text{ as } k \to \infty.
\]

(a) and (b*) assure that $\text{var}(MS_{\text{bet}}(Y_k)) \to 0$ as $k \to \infty$. The proof relies mainly on the expression for the variance of a quadratic contrasts, as found, e.g., in Searle (1971). For the proof of these results can be found in the Appendix.

\(^6\)More general SLLN conditions for the weakly dependent random variables $\tilde{Y}_1^2(k), \ldots, \tilde{Y}_m^2(k)$ can be found in Lyons (1988).
3.4 Evaluating prediction for correlated responses

Although there are many uses for estimating the explainable variance, we focus on its role in assessing prediction models. Roddey et al. (2000) show that explainable variance upper bounds the accuracy of prediction on the sample when noise is iid. We generalize their results for arbitrary noise correlation and account for generalization from sample to population\(^7\).

As shown in Lemma 2, the noise level \(\bar{\sigma}_\epsilon^2\) is the optimal expected loss under mean square prediction error (MSPE) loss, and the explainable variance \(\omega^2\) approximates the accuracy under squared-correlation \(\text{Corr}^2\) utility.

A more explicit notation setup is needed for studying the relation between predictions and signal responses. Let \(f\) be a prediction function that predicts a real-valued response to any possible image \(I\), out of a large population of \(M\) images,

\[ f : \{I_i\}_{i=1}^M \to \mathbb{R}. \tag{3.20} \]

We will assume \(f\) does not depend on the sample we are evaluating, meaning that it was fit on separate data. We usually think of \(f\) as using some aspects of the image to predict the response, although we do not restrict it in any parametric way to the image.

Prediction accuracy is measured only on the \(m\) images sampled for the (non-overlapping) validation set. Let \(s : \{1, ..., m\} \to \{1, ..., M\}\) be the random sampling function, and \(I_s(j)\) the random image chosen for the \(j\)'th sample image.

Furthermore, let us introduce notation relating the random effects to the image sampling. For this, assume each image is associated with a mean activation quantity \(\mu_i\), so that \(\sum \mu_i = 0\) and \(\sum \mu_i^2 = \sigma_A^2\). Then the random effects \(A_j\) defined before can now be described \(A_j = \mu_{s(j)}\).

To evaluate prediction accuracy, the predicted response \(f(I_s(j))\) is compared with the averaged (observed) response for that image \(\bar{Y}_j\). We consider two common accuracy measures: mean squared prediction error (MSPE\([f]\)) and the squared correlation (\(\text{Corr}^2\[f]\)), defined

\[ \text{MSPE}[f] := \frac{1}{m-1} \sum_{j=1}^m (f(I_s(j)) - \bar{Y}_j)^2, \tag{3.21} \]

\[ \text{Corr}^2[f] := \text{Corr}^2\big(f(I_s(j)), \bar{Y}_j\big) = \frac{\left(\frac{1}{m-1} \sum_{j=1}^m (f(I_s(j)) - \bar{f}_s)(\bar{Y}_j - \bar{Y})\right)^2}{\frac{1}{m-1} \sum_{j=1}^m (f(I_s(j)) - \bar{f}_s)^2 \sum_{j=1}^m (\bar{Y}_j - \bar{Y})^2}, \tag{3.22} \]

where \(\bar{f}_s\) denotes the average of the predictions for the sample.

We will state and discuss the results relating the explainable variance to optimal prediction; details can be found in the Appendix.

\(^7\)While these results may have been proved before, we have not found them discussed in similar context.
Lemma 2. Let \( f^*: \{I_i\}_{i=1}^M \rightarrow \mathbb{R} \) be the prediction function that assigns for each stimulus \( I_i \) its mean effect \( \mu_i \), or \( f^*(I_i) = \mu_i \). Under the model described in Section 3.2.3,

(a) \( f^* = \min_f \mathbb{E}_{A,E}[MSPE[f]] \);

(b) \( \sigma^2 = \mathbb{E}_{A,E}[MSPE[f^*]] = \min_f \mathbb{E}_{A,E}[MSPE[f]] \);

(c) \( \omega^2 \approx \mathbb{E}_{A,E}[\text{Corr}^2[f^*]] \) with a bias term smaller than \( \frac{1}{m-1} \).

Under our random effects model, the best prediction (in MSPE) is obtained by the mean effects, or \( f^* \). More important to us, the accuracy measures associated with the optimal prediction \( f^* \) can be approximated by signal and noise levels: \( \sigma^2 \) for \( MSPE[f^*] \) and \( \omega^2 \) for \( \text{Corr}^2[f^*] \).

The main consequence of this lemma is that the researcher does not need a "good" prediction function to estimate the "predictability" of the response. Prediction is upper-bounded by \( \omega^2 \), a quantity which can be estimated without setting a specific function in mind. Moreover, when a researcher does want to evaluate a particular prediction function \( f \), \( \hat{\omega}^2 \) can serve as a yard stick with which \( f \) can be compared. If \( \text{Corr}^2[f] \approx \hat{\omega}^2 \), the prediction error is mostly because of variability in the measurement. Then the best way to improve prediction is to reduce the noise by preprocessing or by increasing the number of repeats. On the other hand, if \( \text{Corr}^2[f] \ll \hat{\omega}^2 \), there is still room for improvement of the prediction function \( f \).

3.5 Simulation

We simulate data with a noise component generated from either a block structure or a timeseries structure, and compute shuffle estimates for signal variance and for explainable variance. For a wide range of signal-to-noise regimes, our method produces unbiased estimators of \( \sigma_A^2 \). These estimators are fairly accurate for sample sizes resembling our image-fMRI data, and the bias in the explainable variance \( \omega^2 \) is small compared to the inherent variability. These results are shown in Figure 3.5. In Figure 3.6 we show that under non-zero \( \sigma_A^2 \), the shuffle estimates have less bias and lower spread compared to the parametric model using the correctly specified noise correlation.

3.5.1 Block structure

For the block structure we assumed the noise is composed of an additive random block effect constant within blocks \( (b_k, k = 1, ..., B \text{ blocks}) \), and an iid Gaussian term \( (e_t, t = 1, ..., T) \)

\[ Y_t = A(t) + b_{\beta(t)} + e_t \]
$A_j, b_k$ and $e_t$ are sampled from centered normal distributions with variances $(\sigma_A^2, \sigma_b^2, \sigma_e^2)$. We used $\sigma_b^2 = 0.5, \sigma_e^2 = 0.7$, and varied the signal level $\sigma_A^2 = 0, 0.1, \ldots, 0.9$. We used $m = 120$, $n = 15$, with all presentations of every 5 stimuli composing a blocks ($B = 20$ blocks). For each of these scenarios we ran 1000 simulations, sampling the signal, block, and error effects. $MS_{bet}$ was estimated the usual way, and $P$ was chosen to be a random permutation within each block ($\alpha = 0.115$). The results are shown in Figure 3.5 a.

**Figure 3.5:** Simulations for the block and time-series (a) Simulation results comparing shuffle estimates for signal variance $\sigma_A^2$ (black) and explainable variance $\omega^2$ (blue) to the true population values (dashed line). Noise correlation followed an independent block structure: noise within blocks was correlated, and between blocks was independent. The x-axis represents the true signal variance $\sigma_A^2$ of the data, and the y-axis marks the average of the estimates and $[0.25,0.75]$ quantile range. (b) A similar plot for data generated under a stationary time-series model.

### 3.5.2 Time-series model

For the time-series model we assumed the noise vector $e \in \mathbb{R}^T$ is distributed as a multivariate Gaussian with mean 0 and a covariance matrix $\Sigma$, where $\Sigma$ is an exponentially decaying covariance with a nugget,

$$
\Sigma_{tu} = \rho_{|t-u|} = \lambda_1 \cdot exp\{-|t - u|/\lambda_2\} + (1 - \lambda_1)1_{(t=u)}.
$$
Figure 3.6: Comparison of methods on simulation. Each pair of box-plots represents the estimated signal variance $\sigma^2_A$ using the shuffle estimator (dark gray) and REML (light gray) for 1000 simulations. The blue horizontal line represents the true value of $\sigma^2_A$. The REML estimator assumes the correct model for the noise, while the shuffle estimator only assumes a stationary time series. When there is no signal, REML outperforms the shuffle estimators, but in all other cases it is both biased and has greater spread.

Then $Y = A(t) + e_t$ with the random effects $A(t)$ sampled from $\mathcal{N}(0, \sigma^2_A)$ for $\sigma^2_A = 0, 0.1, ... 0.9$. We used $m = 120, n = 15$, and the parameters for the noise were $\lambda_1 = 0.7$ and $\lambda_2 = 30$, meaning $\rho_{125} \approx 0.01$. The schedule of treatments was generated randomly. For each of these scenarios we ran 1000 simulations, sampling the signal and the noise. In Figure 3.5 b we estimated the shuffle estimator with $P$ the reverse permutation ($g_P(t) = T + 1 - t$), resulting in $\alpha = 0.064$.

### 3.5.3 Comparison to REML

In Figure 3.6 we used time-series data to compare $\sigma^2_A$ estimates based on the shuffle estimators to those obtained by an REML estimator with the correct parametrization for the noise correlation matrix. We used nlme package in R to fit a repeated measure analysis of variance for the exponentially decaying correlation of noise with a nugget effect. The comparison included 1000 simulations for $\sigma^2_A = 0, 0.2, 0.4, 0.6, 0.8$, and a noise model identical to Section 3.5.2.
3.5.4 Results

Figure 3.5 describes the performance of shuffle estimates on two different scenarios: block correlated noise (a), and stationary time-series noise (b). For signal variance (black) the shuffle estimator gives unbiased estimates. The shuffle estimator for explainable variance is not unbiased, but the bias is negligible compared to the variability in the estimates. In Figure 3.6, we compare the signal variance estimates based on the shuffle estimator (dark gray) with estimates based on REML (light gray). The estimates based on the shuffle have no bias, while those based on REML underestimate the signal. The variance of the REML estimates is slightly larger, due in part to rare events (between 1%-0.5% of runs) in which the estimated signal variance was effectively 0 - perhaps indicating a problem with the optimization.

3.6 Data

We are now ready to evaluate prediction models using the shuffle estimates for explainable variance. Prediction accuracy was measured for encoding models of 1250 voxels within the primary visual cortex (V1). Because V1 is functionally homogenous, encoding models for voxels within this cortical area should work similarly. As observed in Figure 3.2, there is large variation between prediction accuracies for the different voxels. We try to explain this observed variation as a result of variation in the explainable variance. To do this, prediction accuracy values for these 1250 voxels are compared to the explainable variance estimates generated by the shuffle estimator for each voxel. We also compare the accuracy values to alternative estimates for explainable variance, using the method of moments for uncorrelated noise, and REML under several parameterizations for the noise.

3.6.1 Methods

We estimate the explainable variance of voxels \( \omega^2 = \sigma_A^2 / \sigma_Y^2 \) with several different methods. The methods differ in how \( \sigma_A^2 \) is estimated; all methods use the sample averages variance \( MS_{bet}(Y) \) for \( \sigma_Y^2 \), and plug in the two estimates into \( \omega^2 \). We estimate \( \omega^2 \) separately for each voxel \( (r = 1, ..., 1250) \). The methods we compare are

1. The shuffle estimator. We assume the noise follows a stationary time-series model within each block, and is independent between the blocks. We therefore choose a permutation \( P \) that reverses the order of the measurements, \( (PY)_t = Y_{T+1-t} \). Because the size of the blocks is identical, reversing the order of the data vector is equivalent to reversing the order within each block. \( \alpha = 0.14 \). Specifically, the estimator is restricted to be
positive:

$$\hat{\sigma}^2_{A+} = \max \left\{ \frac{MS_{bet}(Y) - MS_{bet}(PY)}{1 - \alpha}, 0 \right\}$$

for signal variance, and $$\hat{\omega}^2 = \hat{\sigma}^2_{A+}/MS_{bet}$$ for the explainable variance.

2. An estimator ($$\hat{\omega}^2$$) unadjusted for correlation. We use the mean-square within ($$MS_{wit} = \frac{1}{(m-1)n} \sum_{j=1}^{m} \sum_{t:h(t)=j}(Y_t - \bar{Y}_j)^2$$) contrast to estimate the noise variance $$\sigma^2$$, scale by $$1/n$$ to estimate the noise level $$\hat{\sigma}^2$$, and remove the scaled estimate from $$MS_{bet}$$,

$$\hat{\sigma}^2_A = MS_{bet} - MS_{wit}/n.$$ 

Explainable variance is obtained by plug in estimator $$\hat{\omega}^2 = \hat{\sigma}^2_A/MS_{bet}$$.

3. Estimators based on a parametric noise model.

- We assume the noise is generated from an exponentially decaying correlation matrix, with a nugget effect. This means

$$\Sigma_{t,t+d} = \lambda_2 e^{x(-d/\lambda_1)} + 1_{(d=0)}(1 - \lambda_2)$$

where the rate of decay $$\lambda_1$$ and nugget effect $$\lambda_2$$ where additional parameters. If $$\lambda_2 = 0$$, this is equivalent to the AR(1) model.

- Alternatively, we assume the noise is generated from an AR(3) process, or $$\epsilon_t = \eta_t + \sum_{k=1}^{3} a_k \epsilon_{t-k}$$. This models allows for non-monotone correlations.

We use the nlme package in R to estimate the signal variance of this model using restricted maximum likelihood (REML, e.g. Laird et al. 1987), and use the plug-in estimator for the explainable variance.

3.6.2 Results

In Figure 3.7 we compare the prediction accuracy of the voxels to estimates of the explainable variance. Each panel has 1250 points representing the 1250 voxels: the x coordinate is the estimate of explainable variance for the voxel, and the y coordinate is $$Corr^2[f]$$ for the Gabor based prediction-rule. The large panel shows the shuffle estimators for explainable variance. The relation between $$y = Corr^2[f]$$ and $$\hat{\omega}^2$$ is very linear ($$r = 0.9$$). The estimated slope and intercepts (using least-squares) for this relation are $$y \approx 0.66 \cdot \hat{\omega}^2 - 0.009$$. Note that almost all voxels for which accuracy is close to random guessing ($$Corr^2[f] < 0.05$$) could be identified based on low explainable variance without knowledge of the specific feature set.

When we try to repeat this analysis with other $$\omega^2$$ estimators, explainable variance estimates are no longer strongly related with the prediction accuracy. When correlation in the noise
is ignored (b), signal strength is greatly overestimated. In particular, some of the voxels for which prediction accuracy is almost 0 have very high estimates of explainable variance (as high as $\tilde{\omega}^2 = 0.8$). In contrast to the shuffle estimates, it is hard to learn from these explainable variance estimates about the prediction accuracy for a voxel.

This incompatibility of prediction accuracy and explainable variance estimates is also observed when the estimates are based on maximum likelihood methods that parameterize the noise matrix. For the AR(3) model in (d), we see variability between explainable variance estimates for voxels with given prediction accuracy level. The smaller model (c) seems to suffer from both over-estimation of signal and from high variance.

Figure 3.7: **Optimal vs. observed prediction accuracy.** The estimated optimal prediction is compared with observed prediction ($\text{Corr}^2$), each point representing a V1 response. The optimal prediction estimated by (a) shuffle estimators accounting for stationary noise distributions; (b) Method of moments estimator assuming independent noise; (c) REML estimator assuming exponential decay of noise with nugget within blocks; and (d) REML estimator assuming an AR(3) model for the noise correlation within blocks. The x=y is plotted in blue.
3.7. DISCUSSION

Figure 3.8: Comparison of voxels from two functional visual areas: V1 and V4. The estimated optimal prediction is compared with observed prediction ($Corr^2$) for voxels from two different visual areas: V1 (black points, as in Figure 3.7), and V4 (red points). Data for all voxels was collected simultaneously, regardless of visual area. The black and red lines represent the linear least-square fit for V1 and V4 voxels respectively. A clear difference is observed between the two visual areas regarding the relation between prediction accuracy to the explainable variance. This indicates that the lower prediction accuracy for V4 voxels can be partly mitigated with better prediction models.

3.7 Discussion

We have presented the shuffle estimator, a resampling-based estimator for the explainable variance in a random-effects additive model with auto-correlated noise. Rather than parameterize and estimate the correlation matrix of the noise, the shuffle estimator treats the contribution of the noise to the total variance as a single parameter. Symmetries in the data-collection process indicate those permutations which, when applied to the original data, would not change the contribution of the noise. An unbiased estimator of the signal variance is derived from differences between the total variance of the original data vector and the shuffled vector. The resulting estimate of signal variance is plugged in as the enumerator for the explainable variance ratio estimate.

For a brain-encoding experiment, we have shown that the strong correlation present in the fMRI measurements greatly compromises classical methods for estimating explainable variance. We used prediction accuracy measures of a well-established parametric model for voxels in the primary visual cortex as indicators of the explainable signal variance at each
of the voxels. Shuffle estimates of the explainable variance explained most of the variation between voxels, even though they were blind to features of the image. Other methods did not do well: methods that ignored noise correlation seem to greatly overestimate the explainable variance, while methods that estimated the full correlation matrix were considerably less informative with regards to prediction accuracy. We consider this convincing evidence that the shuffle estimators for explainable variance can be used reliably even when no gold-standard prediction model is present.

Explainable variance is an assumption-less measure of signal, in that it makes no assumptions about the structure of the mean function that relates the input image to response. We find it attractive that the shuffle estimator for explainable variance similarly requires only weak assumptions for the correlation of the noise. This makes the shuffle estimator a robust tool, which can used at different stages of the processing of an experiment: from optimizing of the experimental protocol, through choosing the feature space for the prediction models, to fitting the prediction models.

Choice of Permutations

The bias and variance of the shuffle estimator depend on the permutation underlying the shuffle. Different permutations pose different assumptions on the noise correlation as well as provide a different mix of treatments corresponding to different \( \alpha \)s. We recommend the permutation be chosen prior to the analysis, based on the expected noise structure and the mixing constants \( \alpha \)s, in order to minimize the risk of data snooping. Optimally, the experiment could be designed so that a specific permutation - perhaps the reverse permutation - will mix treatments well resulting in a low \( \alpha \). In our experiment, the reverse and other regular permutations such as shifts had low \( \alpha \)s because the design was generated using an irregular pseudo-random sequence. Moreover, when several noise-conserving permutations exist and have similar \( \alpha \)s, it may be preferable to average the corresponding shuffle estimators to reduce the variance of the estimators.

In cases were no symmetry permutations are useable, a wider class of "almost noise conserving" permutations can be considered. To give concrete examples, consider the following two permutations: A cyclic left-shift permutation so that \( (P_1 Y)_t = Y_{t+1} \) for \( t = 1, \ldots, T - 1 \), and \( (P_1 Y)_T = Y_1 \); and a permutation of odd and even channels \( (P_2 Y)_{2s-1} = Y_{2s} \), \( (P_2 Y)_{2s} = Y_{2s-1} \), for \( s = 1, \ldots, T/2 \). Neither is an exact symmetry of a covariance matrix that represents a stationary time series. In \( P_1 \), the first measurement in each block is not correlated to the sequence. \( P_2 \) is even farther from symmetry, in that the medium and long range correlations are conserved but the local structure is scrambled.

Nevertheless, the shuffle estimates from either \( P_1 \) or \( P_2 \) produce, when compared to predictions, population results that are similar to those observed for the reverse permutation. The new
estimates compare in both linearity, with \( r = 0.905 \) between \( \text{Corr}^2[f^*] \) and \( \hat{\omega}^2 \) for both \( P_1 \) and \( P_2 \) compared to \( r = 0.9 \) for the reverse permutation, as well in the slope (0.65 for \( P_1 \), 0.68 for \( P_2 \)) of the linear trend and its intercept (0 for both). This does not imply that any permutation would work well; indeed, shuffling with a random permutation, implicitly assuming IID noise, results in a similar bias as that observed in the method of moments estimator shown in Figure 3.7 b. Note that for any candidate \( P \) and possible \( \Sigma \), the degree to which noise is conserved can be explicitly measured by comparing \( tr(\Sigma(B - G)) \) to \( tr(P^T\Sigma P(B - G)) \).

The shuffle estimators may be useful for applications outside of neuroscience. These estimators can be used to estimate the variance associated with the treatments of an experiment, conditioned on the design, whenever measurement noise is correlated. Spatial correlation in measurements arise in many different domains, from agricultural experiments to DNA microarray chips. Shuffle estimators could provide an alternative to parametric fitting of the noise contributions for these applications.

Future research should be directed at expressing the variance of the shuffle estimator for a candidate permutation, as well as at developing optimal ways to combine information from multiple noise conserving permutations. More generally, shuffle estimators are a single example of adapting relatively new non-parametric approaches from hypothesis testing into estimation; we see much room for expanding the use of permutation methods for creating robust estimators for experimental settings.

### 3.8 Appendix

#### 3.8.1 Consistency of \( \omega^2 \)

Consider a sequence of measurement vectors \( Y(k) \in \mathbb{R}^{T(k)} \) of growing size (\( T(k) \to \infty \)), generated by the additive random effects model described before. The distribution of each data vector can be parametrized by the following parameters \( (\sigma_A^2(k), \sigma_Y^2(k), X(k), \Sigma(k)) \), and the total variance would be:

\[
\bar{\sigma}_Y^2(k) := \mathbb{E}[MS_{bet}(Y(k))].
\]

Assume that both the signal and total variances converge

\[
\sigma_A^2(k) \to \sigma_A^2 \quad \text{and} \quad \sigma_Y^2(k) \to \bar{\sigma}_Y^2 > 0.
\]

Further consider a sequence of shuffle estimators based on noise-conserving permutations \( P(k) \) with mixing constants that are bounded away from 1 (\( \alpha(k) \leq \alpha^0 < 1 \) for each \( k \)). For this setting, we will discuss the sufficient conditions for the shuffle estimator to be consistent:

\[
\hat{\omega}^2(k) \xrightarrow{P} \omega^2 = \sigma_A^2/\bar{\sigma}_Y^2.
\]
For real data, the convergence $\bar{\sigma}_2^2(k) \to \bar{\sigma}_2^2$ is not always meaningful. Explainable variance is a non-asymptotic property of the data, as it is not only a function of the treatment and noise variances ($\sigma_A^2, \sigma_\epsilon^2$), but also of the design and noise autocorrelation. Whereas $\sigma_A^2, \sigma_\epsilon^2$ could be stable as the experiment grows, we expect the impact of correlation to (typically) diminish and therefore explainable variance to increase. Nevertheless, consistency would imply the estimator improves with more measurements.

**Proposition 2.** Under the model as described in above and a sequence of experiments described above, assume that both $A(k)X(k)$ and $\epsilon(k)$ are multivariate normals vectors. Then the following conditions assure consistency of $\hat{\omega}_2^2(k)$:

(a) The number of treatments increases with $k$

$$m(k) \to \infty \text{ as } k \to \infty;$$

(b) The largest $m - 1$ eigenvalues $\lambda_{(1)}(k), \ldots, \lambda_{(m-1)}(k)$ of the noise correlation matrix $\Sigma(k)$ satisfy

$$\frac{1}{n^2(m-1)^2} \sum_{i=1}^{m-1} \lambda_{(i)}^2(k) \to 0 \text{ as } k \to \infty.$$

Furthermore, if the permutations $P(k)$ are symmetries ($P(k)\Sigma(k)P^t(k) = \Sigma(k)$ for all $k$), (a) and (b) can be replaced by the following condition ($b^*$):

$$\text{(b*) } \text{var} [MS_{bet}(Y(k))] \to 0 \text{ as } k \to \infty.$$

($b^*$) directly implies (a), but does not imply (b).

**Remarks**

1. Condition ($b^*$) gives an intuitive answer to the consistency question: the shuffle estimator is consistent when the variance of the $MS_{bet}$ statistic decays. In other words, as long as the variance of $\frac{1}{m-1} \sum_j 1^{m(k)}Y_j^2(k) \to 0$ as $k \to \infty$.

2. In particular, the variance of the estimator will not decay if the number of treatments does not increase. The uncertainty due to the sampling needs to be addressed for the variance to decrease.

3. SLLN can be invoked to for non-Gaussian $Y$ (see remark); the benefit in using the Gaussianity is that explicit conditions on $\Sigma$ can be found.

**Proof.** Here is an outline of the proof. The shuffle estimator is a ratio of two unbiased estimators, and therefore is consistent if the variance of both its numerator and denominator

---

Footnote: $\sigma_\epsilon^2$ should not be confused with the noise level $\bar{\sigma}_2^2$ that may well decay as $T$ grows.
converges to 0. With control of the α’s, this requires that the variance of the quadratic contrasts \( MS_{bet}(Y) \) and \( MS_{bet}(PY) \) would diminish.

To analyze these variances, we use a quadratic contrast identity for multivariate gaussian vectors. The number of treatments \( m \) needs to increase with \( k \) (a) for the signal part of the variance to diminish. A condition on the decay of the noise (b) assures convergence of the noise component in either the regular or the permuted contrasts. Finally, we show that for symmetries \( var(MS_{bet}(PY)) \leq var(MS_{bet}(Y)) \), in which case (b*) can replace (a) and (b).

**A ratio estimator**

The shuffle estimator is a ratio of two unbiased estimators

\[
\hat{\omega}^2 = \frac{\hat{\sigma}_A^2}{\hat{\sigma}_Y^2}
\]

Convergence in probability of \( a(k)/b(k) \rightarrow a/b \) is assured if \( a(k) \overset{P}{\rightarrow} a \) and \( b(k) \overset{P}{\rightarrow} b > 0 \). Furthermore, an unbiased estimator converges in probability if its variance converges to 0. We will therefore prove below that under the specified conditions both \( var(\hat{\sigma}_A^2(k)) \) and \( var(\hat{\sigma}_Y^2(k)) \) converge to 0.

Let us then take a closer look at the estimators for the ratio factors

\[
\hat{\sigma}_A^2 = \frac{MS_{bet}(Y) - MS_{bet}(PY)}{1 - \alpha}; \\
\hat{\sigma}_Y^2 = MS_{bet}(Y).
\]

The variance of the enumerator expands to:

\[
var[\hat{\sigma}_A^2] = \frac{1}{(1 - \alpha)^2} \left( var[MS_{bet}(Y)] + var[MS_{bet}(PY)] - 2cov(MS_{bet}(Y), MS_{bet}(PY)) \right),
\]

with \( P, \alpha, \Sigma = E e[YY'] \), and \( B = XX'/n \) (the averaging matrix for \( MS_{bet} \)) all depending on \( k \). The factor \((1 - \alpha)^{-2} \leq (1 - \alpha^0)^{-2} \) for all \( k \). It therefore suffices to show that \( var(MS_{bet}(Y)) \) and \( var(MS_{bet}(PY)) \) to 0 for the variance of the enumerator to converge: regardless its sign, the covariance will diminish if the variances of both expressions converge to 0. Finally, the denominator is also \( \hat{\sigma}_Y^2 = MS_{bet}(Y) \), so the above convergences would conclude the proof.

**Variance of quadratic contrasts**

The key step is to express \( var(MS_{bet}(Y)) \) and \( var(MS_{bet}(PY)) \) as variances of the quadratic contrasts \( \frac{1}{n(m-1)}Y'(B - G)Y \) and \( \frac{1}{n(m-1)}Y'[P'(B - G)P]Y \) respectively. We use the following known result (e.g. Searle (1971), page 55):

---

\(^9\)We will drop references to \( k \) from now.
For a generic multivariate normal \( \mathbf{Y} \) with \( E[\mathbf{Y}] = 0, \ E[\mathbf{YY}'] = \mathbf{V} \), and a symmetric matrix \( \mathbf{Q} \),
\[
\text{var}(\mathbf{Y}'\mathbf{Q}\mathbf{Y}) = 2\text{tr}(\mathbf{Q}\mathbf{V}\mathbf{Q}) = 2\text{tr}(\mathbf{Q}\mathbf{V}^2).
\]

For \( \mathbf{Y} = \mathbf{A}\mathbf{X} + \epsilon \) with \( E[\mathbf{Y}] = 0 \) and \( E[\mathbf{YY}'] = \mathbf{V} = \sigma^2_A\mathbf{B} + \sigma^2_\epsilon\mathbf{\Sigma} \), this becomes
\[
\text{var}(\text{MS}_{\text{bet}}(\mathbf{Y})) = 2\text{tr}\left(V\frac{1}{n(m-1)}(\mathbf{B} - \mathbf{G})V\frac{1}{n(m-1)}(\mathbf{B} - \mathbf{G})\right),
\]
which can be expanded as follows:
\[
\text{var}(\text{MS}_{\text{bet}}(\mathbf{Y})) = \frac{2}{n^2(m-1)^2}\text{tr}\left(\left(\sigma^2_A\mathbf{B} + \sigma^2_\epsilon\mathbf{\Sigma}\right)(\mathbf{B} - \mathbf{G})\left(\sigma^2_A\mathbf{B} + \sigma^2_\epsilon\mathbf{\Sigma}\right)(\mathbf{B} - \mathbf{G})\right)
\]
\[
= \frac{2}{n^2(m-1)^2}\text{tr}\left[\sigma^2_A\mathbf{B}((\mathbf{B} - \mathbf{G})\sigma^2_A\mathbf{B}(\mathbf{B} - \mathbf{G})\right] \quad [\text{SIG}]
\]
\[
+ \sigma^2_\epsilon\mathbf{\Sigma}(\mathbf{B} - \mathbf{G})\sigma^2_\epsilon\mathbf{\Sigma}(\mathbf{B} - \mathbf{G}) \quad [\text{NOI}]
\]
\[
+ 2\sigma^2_A\mathbf{B}(\mathbf{B} - \mathbf{G})(\sigma^2_\epsilon\mathbf{\Sigma})(\mathbf{B} - \mathbf{G}) \quad [\text{INT}].
\]

The three variance components correspond to the contributions of signal (SIG), of noise (NOI), and of their interaction (INT).

**Convergence of signal and interaction components of** \( \text{var}(\text{MS}_{\text{bet}}\mathbf{Y}) \).  
SIG and INT diminish iff the number of treatments \( m \) increases to infinity (a). Recall that \( \mathbf{B} \) and \( (\mathbf{B} - \mathbf{G}) \) are projections, with \( (\mathbf{B} - \mathbf{G})^2 = \mathbf{B}(\mathbf{B} - \mathbf{G}) = \mathbf{B} - \mathbf{G} \), and \( \text{tr}(\mathbf{B} - \mathbf{G}) = \text{rank}(\mathbf{B} - \mathbf{G}) = m - 1 \). The signal component therefore simplifies to
\[
\text{SIG} = \frac{2n^2\sigma^4_A}{n^2(m-1)^2}\text{tr}(\mathbf{B} - \mathbf{G}) = \frac{2\sigma^4_A}{m-1}.
\]

This reflects that the sample variance converges to the population variance only as \( m \) increases.  
Similarly, the interaction component simplifies to
\[
\text{INT} = \frac{4n\sigma^2_A\sigma^2_\epsilon}{n^2(m-1)^2}\text{tr}((\mathbf{B} - \mathbf{G})\mathbf{\Sigma}).
\]

Because \( \sigma^2_\epsilon = \sigma^2\frac{n}{m-1}\text{tr}((\mathbf{B} - \mathbf{G})\mathbf{\Sigma}) \),
\[
\text{INT} = \frac{4\sigma^2_A\sigma^2_\epsilon}{m-1},
\]

which also diminishes as \( m \to \infty \).

**Convergence of the noise component.**  
We control the noise component more crudely with (b), which assumes that the energy of \( \mathbf{\Sigma} \) becomes less concentrated as \( \mathbf{\Sigma} \) grows. Because \( (\mathbf{B} - \mathbf{G}) \) is an idempotent matrix of rank
m − 1, (B − G) projects Σ onto the m − 1 eigen-directions of (B − G). The eigenvalues of the projection Σ(B − G) are controlled by the highest m − 1 eigenvalues of Σ. More precisely,

\[ tr \left( ((B - G) \Sigma)^2 \right) = \sum_{i=1}^{m-1} \eta^2_i \leq \sum_{i=1}^{m-1} \lambda^2_i \]

where \( \eta_i \) and \( \lambda_i \) are the \( i \)th largest eigenvalues of \( \Sigma(B - G) \) and \( \Sigma \) respectively. (b) implies that \( \frac{1}{n^2(m-1)^2} tr \left( ((B - G) \Sigma)^2 \right) \to 0. \)

**Convergence of \( MS_{bet}(PY) \)**

The variance of the \( MS_{bet} \) of the permuted vector can be similarly controlled by (a) and (b). We can decompose it as we did \( MS_{bet}(Y) \):

\[ var(MS_{bet}(PY)) = SIG_P + INT_P + NOI_P \]

\[ tr(B(B_P - G)) \leq tr(B(B - G)) \] from Cauchy-Schwartz inequality, and \( tr((B - G)\Sigma) = tr((B_P - G)\Sigma) \) from the definition of noise-conserving permutations. Hence, \( SIG_P \leq SIG, INT_P \leq INT \), meaning that both \( SIG_P \) and \( INT_P \) decay as \( m \to \infty \).

Furthermore, the permuted version of \( B, B_P := P'BP \), is still a projection of rank \( m - 1 \), whereas \( G \) is invariant to permutations. \( NOI_P \) therefore decays with (b) by controlling the \( m - 1 \) top eigenvalues.

**Condition b*.**

If \( P \) is a symmetry of the \( \Sigma \), meaning \( (B_P - G)\Sigma = (B - G)\Sigma \) as matrices, then \( tr(((B_P - G)\Sigma)^2) = tr(((B_P - G)\Sigma)^2) \) and \( NOS = NOS_P \). We conclude that in that case \( var(MS_{bet}(PY)) \leq var(MS_{bet}(Y)) \), so \( (b^*) \) is sufficient to finish the proof.

\[ \square \]

**3.8.2 Lemma (2)**

Let \( f^* : \{I_i\}_{i=1}^M \to \mathbb{R} \) be the prediction function that assigns for each stimulus \( I_i \) its mean effect \( \mu_i \), or \( f^*(I_i) = \mu_i \). Under the experimental conditions and model described above,

(a) \( f^* = \arg \min_f \mathbb{E}_{A,s}[MSPE[f]] \);

\footnote{(b) is too strong because it does not take into account the fact that \( B - G \) removes the global average. In cases where the noise is uniformly correlated, the dominating eigenvector would typically be removed by \( G \).}
(b) \( \bar{\sigma}^2 = \mathbb{E}_{A_\epsilon}[MSPE[f^*]] = \min_f \mathbb{E}_{A_\epsilon}[MSPE[f]] \):

(c) \( \omega^2 \approx \mathbb{E}_{A_\epsilon}[Corr^2[f^*]] \) with a bias term smaller than \( \frac{1}{m-1} \).

Proof. (a) + (b)

First, for any image in the sample, we compare the prediction with the expected average given the sampling,

\[
MSPE[f] = \frac{1}{m-1} \sum_j \left( (f(I_{s(j)}) - \mathbb{E}_e[\bar{Y}_j]) + (\mathbb{E}_e[\bar{Y}_j] - \bar{Y}_j) \right)^2. \tag{3.23}
\]

From our model, \( \mathbb{E}_e[\bar{Y}_j] = A_j \). Substituting this into 3.23 and taking expectation over the noise, we get

\[
\mathbb{E}_e[MSPE[f]] = \frac{1}{m-1} \mathbb{E}_e \left[ \sum_j \left( f(I_{s(j)}) - A_j \right)^2 + \mathbb{E}_e \sum_j (A_j - \bar{Y}_j)^2 \right. \\
\left. + \mathbb{E}_e \sum_j (f(I_{s(j)}) - A_j) (A_j - \bar{Y}_j) \right].
\]

Recall that \( \bar{Y}_j - A_j = \bar{\epsilon}_j \) for each \( j \), with \( \mathbb{E}_e[\bar{\epsilon}_j] = 0 \) and a sample variance \( \bar{\sigma}_\epsilon^2 \). Therefore

\[
\mathbb{E}_e[MSPE[f]] = \frac{1}{m-1} \left[ \sum_j \left( f(I_{s(j)}) - A_j \right)^2 \right] + \bar{\sigma}_\epsilon^2. \tag{3.24}
\]

By also taking an expectation over the sampling

\[
\mathbb{E}[MSPE[f]] = \mathbb{E}_A \mathbb{E}_e[MSPE[f]] = \frac{m}{m-1} \frac{1}{M} \sum_i \left[ (f(I_i) - \mu_i)^2 \right] + \bar{\sigma}_\epsilon^2. \tag{3.25}
\]

\( \square \)

Proof. (c)

Since the optimal \( f^* \) maps each image \( I_i \) to its mean-effect \( \mu_i \), for the sampled image it maps the random effect:

\[
f^*(I_{s(j)}) = \mu_{s(j)} = A_j.
\]

Hence \( Corr^2[f^*] = Corr^2(A_j, \bar{Y}_j) \), or in extended form

\[
Corr^2_j(A_j, \bar{Y}_j) = \frac{\left( \frac{1}{m-1} \sum_{j=1}^m (A_j - \bar{A})(\bar{Y}_j - \bar{Y}) \right)^2}{\left( \frac{1}{m-1} \sum_{j=1}^m (A_j - \bar{A})^2 \right) \left( \frac{1}{m-1} \sum_{j=1}^m (\bar{Y}_j - \bar{Y})^2 \right)} \tag{3.26}
\].
3.8. APPENDIX

Recall that \( \bar{Y}_j = A_j + \bar{\epsilon}_j \). Equation (3.26) becomes

\[
\left( \frac{1}{m-1} \sum_{j=1}^{m} (A_j - \bar{A})(\bar{A}_j - \bar{A}) + (A_j - \bar{A})(\bar{\epsilon}_j - \bar{\epsilon}) \right)^2 \\
\left( \frac{1}{m-1} \sum_{j=1}^{m} (A_j - \bar{A})^2 \right) \left( \frac{1}{m-1} \sum_{j=1}^{m} (\bar{Y}_j - \bar{Y})^2 \right).
\]

(3.27)

Let

\[
s_A^2 = \frac{1}{m-1} \sum_{j=1}^{m} (A_j - \bar{A})^2; \quad \bar{s}_\epsilon^2 = \frac{1}{m-1} \sum_{j=1}^{m} (A_j - \bar{A})^2; \quad MS_{\text{bet}} = \frac{1}{m-1} \sum_{j=1}^{m} (\bar{Y}_j - \bar{Y})^2,
\]

represent the sample variance of the treatment effects, averaged noise, and average measurements respectively. Moreover, let

\[
r = \frac{\sum_{j=1}^{m} (A_j - \bar{A})(\bar{\epsilon}_j - \bar{\epsilon})}{(m-1) s_A \cdot \bar{s}_\epsilon}
\]

be the empirical correlation of the treatment effects and the averaged noise.

Substituting into Equation 3.27 results in:

\[
\frac{(s_A^2 + s_A \bar{s}_\epsilon r)^2}{s_A^2 MS_{\text{bet}}} = \frac{s_A^2 + 2s_A \bar{s}_\epsilon r + \bar{s}_\epsilon^2 r^2}{MS_{\text{bet}}}.
\]

By taking expectations over \( A \) and \( \epsilon \) and approximating the expectation of the ratio with the ratio of the expectations, we get:

\[
E_{A,\epsilon}[\text{Corr}^2[f^*]] = E_{A,\epsilon} \left[ \frac{s_A^2 + 2s_A \bar{s}_\epsilon r + \bar{s}_\epsilon^2 r^2}{MS_{\text{bet}}} \right] \\
= E_{A,\epsilon} \left[ \frac{s_A^2}{MS_{\text{bet}}} \right] + E_{A,\epsilon} \left[ \frac{2s_A \bar{s}_\epsilon r + \bar{s}_\epsilon^2 r^2}{MS_{\text{bet}}} \right] \\
\approx \frac{E_{A,\epsilon}[s_A^2]}{E_{A,\epsilon}[MS_{\text{bet}}]} + \frac{E_{A,\epsilon}[2s_A \bar{s}_\epsilon r + \bar{s}_\epsilon^2 r^2]}{E_{A,\epsilon}[MS_{\text{bet}}]} \\
= \frac{\omega^2 + E_{A,\epsilon}[2s_A \bar{s}_\epsilon r + \bar{s}_\epsilon^2 r^2]}{\sigma_Y^2}.
\]

Since the mean effects \( A_j \)'s and averaged noise \( \bar{\epsilon}_j \)'s are independent, \( E[s_A \bar{s}_\epsilon r] = 0 \). Hence

\[
E_{A,\epsilon}[\text{Corr}^2[f^*]] \approx \omega^2 + \frac{E[r^2]}{\sigma_Y^2}.
\]

Under mild conditions and \( m \) large enough \( \sqrt{m-1} \approx N(0,1) \). We get a bias on the order of \( \frac{\sigma_Y^2}{\sigma_Y^2} \). Note that unless \( \sigma_Y^2/\sigma_Y^2 \approx 0 \), the bias is negligible compared to the deviation of \( \frac{2s_A \bar{s}_\epsilon r}{MS_{\text{bet}}} \) which is of order \( \frac{1}{\sqrt{m-1}} \).
Chapter 4

Modeling V4 under Naturalistic Conditions with Invariant Image Representations

This chapter presents a full cycle of scientific investigation using prediction models in neuroscience. Whereas the previous chapters developed metrics for evaluating feature sets and prediction models, this chapter takes an extra leap: we use optimization algorithms together with prior scientific knowledge to propose a new image representation. We then fit regularized linear models based on this representation that generalize well to a validation data set. Finally, novel visualization and model-summary techniques help interpret the resulting prediction models, revealing rich patterns of activity in the different neurons and unexpected categories of neurons.

We study responses of neurons from visual area V4 to natural images. V4 is believed to play an important role in the recognition of shapes and objects and in visual attention, although its primary role remains poorly understood. In Chapter 3, we saw that a Gabor-based representation does not predict V4 voxels as well as it does V1 voxels with comparable signal levels. We therefore set out to develop a new image representation, based on invariance and sparse coding principles, that will allow better predictions than the Gabor representation. The resulting feature set is subsequently used to fit prediction models with low-rank approximation for single neuron responses, assuming a redundancy in the spatial-temporal response patterns of the neurons. The linear models based on this non-supervised feature set achieve prediction levels in V4 comparable to the better understood visual areas V1 and V2.

Furthermore, we develop methods to overcome the challenges of analyzing dense, high-dimensional linear models based on an unsupervised feature set. In particular, interpreting patterns across the 2048 different features requires an annotation, as well as statistics that
would summarize the models and account for high correlation between the features. We
conclude that the V4 neuron population prominently splits into two categories: neurons
selective to texture versus those selective to contours, suggesting that one primary role of V4
is to extract image characteristics discriminating objects from background. Moreover, our
study confirms that V4 neurons have diverse selectivities to other features such as orientation,
curvature, and complex shapes such as bars, corners, blobs. This is joint work with Julien
Marial, Ben Willmore, Michael Oliver, Jack Gallant and Bin Yu.

4.1 Introduction

Visual cortex area V4 is located in the mammalian ventral visual pathway. It is believed to
play an important role in the recognition of shapes and objects and in visual attention (Roe
et al. 2012), but its complexity makes it hard to analyze. The primary role of V4 neurons
remains an open question, and its functional organization is not well understood (Roe et al.
2012). Moreover, no current model of V4 has shown good predictions for neuronal responses
to natural images similar to that for V1 and V2 neurons (David and Gallant 2005; Willmore
et al. 2010). A large part of our knowledge about V4 comes from two types of studies. The first type
consists of measuring the effect of V4 cortical lesions in humans (Gallant 2000) and non-human
primates (De Weerd et al. 1996; Schiller 1995; Schiller and Lee 1991). In the second type of
study, a subject is stimulated with images and shapes while collecting electrophysiological
or functional magnetic resonance imaging (fMRI) recordings (Desimone and Schein 1987;
Gallant et al. 1993, 1996; Kobatake and Tanaka 1994; Kotake et al. 2009; Pasupathy and
Connor 1999b, 2002). Even though no simple functional role for V4 has emerged from these
studies, subjects with V4 lesions have been reported with impaired abilities for particular
pattern and shape recognition tasks, illustrating the importance of V4 in shape processing.
V4 neurons have also been shown to be more complex and diverse than those of earlier areas
V1 and V2. For example, receptive fields in V4 are significantly larger than in those in V1
and V2 (Gattass et al. 1988), and V4 neurons exhibit more invariant properties, notably to
the relative stimulus position within the receptive field (Desimone and Schein 1987; Gallant
et al. 1996). Individual cells in V4 have also been shown to be selective to complex features
(Kobatake and Tanaka 1994) and to many visual stimuli characteristics such as orientation,
curvature, motion, color, texture and depth. See (Roe et al. 2012) for a recent extensive
review.

It is hard to infer the behavior of V4 neurons during natural vision from experiments with
artificial stimuli. First, it is not clear that individual V4 cells would similarly respond to
synthetic (gratings, simple shapes) as they do to natural images. Second, neuronal processing
is highly nonlinear: even though a natural scene could probably be decomposed into simple
components, the response to the full scene would be hard to infer from the response to the individual components. It is therefore widely acknowledged that studies involving naturalistic stimuli are crucial for better understanding neural mechanisms involved in cortical visual areas (David et al. 2004; Olshausen and Field 2005; Rust and Movshon 2005; Stanley et al. 2008). Much effort has been made in this direction in V1, V2 and MT studies with success (David and Gallant 2005; Felsen et al. 2005; Nishimoto and Gallant 2011; Sharpee et al. 2006; Willmore et al. 2010). However, modeling V4 neurons has proven more challenging and little is known on the response of V4 neurons to natural scenes.

In this study, we intended to fill in this gap by studying V4 neurons under naturalistic conditions with electrophysiological data. We probed responses of well isolated V4 neurons from two awake macaques that were shown a sequence of grayscale natural images (without color and motion content). We then built a computational model for V4 to overcome the limitations of classical approaches based on Gabor filters. The latter have indeed been successful in predicting the activity of V1 and V2 cells, but do not achieve all types of invariance and selectivity to complex features that V4 neurons exhibit. Our proposed model is based on invariance and sparse coding principles and low-rank regularization, and does not suffer from such limitations. It achieves comparable prediction performance for V4 cells as those reported in the past for V1 and V2 on responses to natural images, despite the fact that V4 is intrinsically more challenging to model (David et al. 2006).

After labeling and categorizing the different image features learned by our computational model, we used sparse principal component analysis (Zou et al. 2006) to interpret the main directions of variation among the observed neuron population. As a result, we discovered as our main finding two groups of neurons: those selective to texture versus those selective to contours. This supports the thesis that one primary role of V4 is to select discriminative image features to extract objects from background, as suggested by the review paper of Roe et al. (2012). Moreover, our study confirms the diversity of V4 neurons, which are selective to orientation, texture, and/or to complex shape features such as bars, corners, and curves.

4.2 Results

We present in this chapter the first analysis of a V4 neuron population using natural images. We probed the activities of 71 neurons from area V4 of two awake macaques who were shown a sequence of natural images while performing a fixation task, and built a computational model to predict the neuron responses and interpret their selectivity profiles.

We started by learning features adapted to natural images using sparse coding (Olshausen and Field 1996) and invariant representations (Cadieu et al. 2007; Lazebnik et al. 2006; Lowe 1999). Then, we built a spatially invariant model based on these adaptive learned
4.2. RESULTS

Such an approach enjoys several benefits over classical image representations where
the features are pre-defined (for example Gabor functions). The set of features we learn is
much richer than classical pre-defined Gabor filters. We demonstrate in this chapter that
our fitted model to V4 responses based on these learned features is easy to interpret, and
can be used to understand the selectivity pattern of each individual neuron. Moreover, the
prediction performance of our fitted model using natural stimuli significantly outperforms
Gabor-based image models, and is similar to that reported in previous studies on V1 and V2
(David and Gallant 2005; Willmore et al. 2010). To the best of our knowledge, this has not
been achieved before by other computational models for V4 (Roe et al. 2012).

4.2.1 Hierarchical Image Feature Extraction for V4 Based on Invariance and Sparse Coding Principles

We briefly describe here how our computational model processes an image, inspired by
successful approaches to scene analysis and object recognition in computer vision (Boureau
et al. 2010; Lazebnik et al. 2006; Lowe 1999). Detailed references to these works and other
models in neuroscience are provided in the discussion section. As far as we know, this
is the first time that such ideas are successfully employed for predicting and interpreting
neurophysiological data.

We present in Figure 4.1 our hierarchical image processing pipeline, which consists of three
successive layers (top, left to right). Each layer is connected to the output of the preceding
one and produces several spatial maps. A map is a two-dimensional representation of an
image after some non-linear transformation. For example, the first layer (top-left) directly
processes an input image and produces 8 orientation maps. Each pixel on these maps is
associated to a specific region in the input image, and each map corresponds to a particular
orientation. This is achieved by keeping the positive response to the image convolution with
8 local oriented filters. Then the orientation maps are smoothed and down-sampled by a
factor 8, introducing invariance for local shifts and deformations.

The second layer is connected to the orientation maps and produces feature maps (center,
magnified in the bottom part of Figure 4.1). They differ from orientation maps in that
they are selective to higher-level features tuned to natural images (e.g. features representing
dges, bars, corners, curves, and texture), and are invariant to local contrast changes. Multi-
orientation patches, each a concatenation of a 4x4 patch in all 8 orientation maps, are
extracted in an overlapping fashion from the orientation maps. They are normalized for
contrast. We then use a linear combination of 2048 high-level features in a previously learned
dictionary to sparsely represent each multi-orientation patch. The linear weights become
pixels in the feature map - each map corresponding to one particular dictionary element.
The dictionary of high-level features was learned in advance, optimized to sparsely represent
Figure 4.1: Hierarchical image processing in our computational model. The figure shows how our computational model processes an image stimulus. A natural image (top-left) is provided as an input to the first layer, which computes orientation maps by convolving the input image with local oriented filters and keeping the positive responses (rectification). The orientation maps are down-sampled. The second layer (expanded on bottom) decomposes these maps into multi-orientation patches that are contrast normalized and sparsely encoded with a dictionary tuned to natural images. The last layer performs non-linear feature pooling inside different pooling regions and produces a high dimensional vector that represents the input stimuli with some invariance properties.

a database of multi-orientation patches sampled from the training image set. A detailed
4.2. RESULTS

A description of the resulting features is given in the next sub-section.

The third layer (top-right) makes the feature maps of the second layer shift invariant inside some pre-defined pooling regions. Each feature map is partitioned into one central and four peripheral areas (see Figure 4.1), and the pixels in a feature map falling into a pooling region are summarized by their square root energy.

The output of this hierarchical image processing pipeline is a high dimensional vector (5x2048) representation for each natural image, with both selectivity and invariant properties. The methodological section provides more technical details.

4.2.2 Dictionary Learning on Orientation Map Patches from Natural Images Produces Complex Interpretable Features

Olshausen and Field (1996) introduced a formulation for learning an overcomplete basis set for natural image patches based on sparse coding principles. Interestingly, they showed that their method automatically produced localized oriented (Gabor-like) image features akin to V1 receptive fields. We applied in our study the same formulation but to multi-orientation patches concatenated from orientation maps, instead of patches from raw natural images. As an empirical result, we automatically learned a set of 2048 dictionary elements that represent more complex features than the Gabor-like features from Olshausen and Field (1996).

A natural question is what kind of shape or surface property each of the learned dictionary element represents? This cannot be answered automatically and requires visualization, interpretation and manual annotation. We thus developed visualization tools to study and categorize these dictionary elements. Since the dictionary elements were not learned on raw image-patches but on multi-orientation patches, they cannot be directly displayed. As detailed in the methodological section, we overcame this difficulty by forming two million pairs of a multi-orientation patch and its corresponding raw image patch. To visualize a feature or a dictionary element, we found the closest multi-orientation patches to this element and displayed their corresponding raw image patches. In Figure 4.2A, we visualize 12 dictionary elements (rows). For each of them, we display (right) the natural image patches from our database whose orientation maps are the most correlated with the element, and (left) their mean.

With a visualization tool in hand, we manually annotated the dictionary elements. We observed that many dictionary elements were associated to image patches from the database with straight edges between bright and dark surfaces (Feature 5 in 4.2A). Other elements were similarly associated to patches with curved edges (Feature 4) of varying complexity, as well as corners (Feature 10). The full taxonomy of dictionary elements we obtained from our visualization includes additional shape features such as bars (Features 1, 6), blobs (Feature
Figure 4.2: **Visualization of Complex Features.** (A) Visualization of 12 dictionary elements. For every element, we identify the 50 natural image patches of size 32 x 32 that, among a database of two million, best match the feature. We display on the left the mean of these matches and display on the right several of them. The mean representation captures the shape encoded by the feature, whereas the patches demonstrate the variation that can be represented by the feature. (B) Feature categories. We display for some feature categories the mean representation of a few dictionary elements. See main text for details.

7, 12), and surface features such as stripes (Feature 3) and non-oriented texture (Feature...
4.2. RESULTS

11). Figure 4.2B provides a broad overview on the visualization of the dictionary elements, categorized by the main shape/surface properties that they correspond to. For downstream analysis, we consolidated the visualization patches (and hence the dictionary elements) into eight categories. Image patches correspond to a feature or element shared not only a shape, but also other image attributes. Different features preferred textured versus un-textured surfaces (compare Feature 2 with Feature 5 in 4.2A). Furthermore, shapes varied in their predominant orientation and in their size (compared to the size of a patch).

Obtaining such a rich set of annotated dictionary elements, which goes beyond classical Gabor filters, is particularly interesting for interpretation purposes. Neuron selectivity profiles that one can discover by interpreting a model are often restricted by the image feature set the model is built on. In contrast, the diversity of our obtained features or dictionary elements translates into large diversity of selectivity profiles our model can characterize.

4.2.3 Prediction Performance in V4 Is as Good as in Earlier Areas V1 and V2.

Not only does adaptive sparse coding give rise to interpretable complex shape and surface features, it is the basis of our state-of-the art prediction models for V4 neurons. We recorded neurophysiological neural activities from 71 well-isolated neurons in area V4 from two awake, behaving, macaques (see data collection section). We probed neuronal behavior using sequences of natural images cropped from grayscale photographs. The sequences of 4,000-12,000 images were presented at 30Hz, centered on the central receptive field (CRF), while the animals fixated. The number of spikes for each neuron was measured at 60Hz, a similar rate as in recent studies about V1 and V2 (Willmore et al. 2010), resulting in two measurements per image. This data was used to learn the parameters of our model. To measure the prediction accuracy, we also acquired a second test data with higher signal to noise ratio, using a sequence of 300 images that were kept aside from the training set.

We simulated a neuron response by following the scheme presented in Figure 4.3. Each image from the sequence was encoded into a feature vector using the three-layer model of Figure 4.1. Then, the response of a neuron at time \( t \) was predicted from the invariant representations of images seen at time lags \( t - 1(16.7ms), t - 2(33.3ms), \ldots, t - 9(166.7ms) \). We fit linear prediction models with low-rank regularization to the observed firing rates of each individual neuron, and evaluated these models on the separate testing data.

Figure 4.4A shows the predictions generated from the model of a single neuron for a novel sequence of test images. These predictions, shown in blue, compare well to the observed spike averages for the same sequence, dashed red, and the correlation between the sequences is 0.67. The prediction accuracy for the neuron population is summarized in 4.4B and measured in correlation coefficients, as often done in the literature (David et al. 2004; Nishimoto and
Figure 4.3: **Predictive model of neuron responses via low-rank regularization.** Input images from a sequence are encoded using the (nonlinear) computational model presented in Figure 4.2. The response of a neuron at a time $t$ is predicted by a linear model involving a few previously seen images - that is, images seen at times $t-1, t-2, \ldots, t-\tau$.

The correlation coefficients are spread between 0.2 and 0.8 with a mean value close to 0.46. The model accounts for, on average, 29% of the explainable variance (sd = 15%). Despite the increased complexity of V4 neurons, these rates are comparable to those obtained in recent studies about V1 and V2 (David and Gallant 2005; Willmore et al. 2010).

A natural question is whether our invariant image representation improves over dense linear combinations of Gabor filters that have been successfully used in the past for modeling V1, V2 and MT cells (Nishimoto and Gallant 2011). Those models consist of the convolution of the input image with a filter bank, followed by rectification and compressive non-linearity steps. We used a static version of the Gabor model developed by Nishimoto and Gallant (2011) using the software package STRFLab v1.45. All parameters of this Gabor model are given in the supplementary material.

We compare our model’s prediction scores with the Gabor baseline and show the results in Figure 4(C). The comparison shows significant improvements of our model (20% increase in correlation on average) over the Gabor baseline ($p$ – value $< 1e^{-10}$ obtained from a paired two-sided Wilcoxon signed rank test).
Figure 4.4: Predictive Performance of Our Model on a Test Set. (A) The prediction and observed responses of the test set for one neuron. The horizontal axis represents the time and the vertical axis the amplitude of the response (centered to zero and normalized). The blue curve represents the mean response observed over time, and the red dashed curve the predicted response. Both are plotted at 60Hz. Prediction accuracy is summarized to be the correlation between the predicted and observed sequences, 0.67 for this neuron. (B): Summary of prediction accuracy across the neuron population in terms of correlation between predicted and true neuron responses. The average correlation is 0.457. (C): a comparison between accuracy of predictions from our model (shown in B) and predictions from a model that uses a dense linear combination of predefined Gabor energy features, which achieves state-of-the-art predictive performance for V1 and V2 neurons. Each point represents correlation scores for a single neuron under the two models.
4.2. RESULTS

Table 4.1: Classification of feature (counts). Each feature is manually assigned into 4 of 19 possible categories, coding the (1) shape, (2) scale, (3) presence of texture, and (4) predominant orientation of the feature. Here we display count co-occurrence of each the shape categories with each of the other categories, for the 2048 features.

<table>
<thead>
<tr>
<th>Shape</th>
<th>Scale</th>
<th>Textures</th>
<th>Orient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
<td>Med</td>
<td>Large</td>
</tr>
<tr>
<td>Straight Edge</td>
<td>12</td>
<td>27</td>
<td>164</td>
</tr>
<tr>
<td>Bar</td>
<td>123</td>
<td>92</td>
<td>68</td>
</tr>
<tr>
<td>Stripes</td>
<td>33</td>
<td>73</td>
<td>54</td>
</tr>
<tr>
<td>Curve/Corner</td>
<td>26</td>
<td>50</td>
<td>172</td>
</tr>
<tr>
<td>Acute/Cut</td>
<td>146</td>
<td>62</td>
<td>53</td>
</tr>
<tr>
<td>Blob</td>
<td>48</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>Multiple Blobs</td>
<td>83</td>
<td>282</td>
<td>113</td>
</tr>
<tr>
<td>Other</td>
<td>2</td>
<td>62</td>
<td>263</td>
</tr>
</tbody>
</table>

4.2.4 Individual Neurons in V4 Respond to Diverse Stimuli Characteristics

The neurons simulated by our models show complex and diverse selectivity profiles. We study the fitted models using two complementary approaches, which either is (a) highlighting the feature categories that strongly excite or inhibit the neuron, or is (b) identifying the images that evoke the strongest excitation according to the model. Each feature was assigned manually, by the first two authors, into one of 8 shape categories; one of five orientation categories (4 directions or non-oriented); one of three scale categories (small, medium, large); and one of three texture categories (weak, mixed, strong). See Table 4.1 for feature-assignment counts. This classification system is described in the methodological section.

For each category we defined a single impact value (see also the methodological section) summarizing the linear coefficients for all features associated with the category. The impact value measured whether the features consistently and effectively excite the neuron (large positive value), inhibit it (large negative), or neither. A value near zero either means that the coefficients were small, or were in conflicting signs. In addition, we identified and displayed the images evoking the most excitatory response according to the model. Impact values were computed after identifying the peak activation lag of the neuron, that is, the delay between the visualization of an excitatory stimuli and the associated neuronal response (see methodological section, and Figure 4.5A for an example).

In the following we comment on four high scoring neurons with different selectivity profiles.
4.2. RESULTS

Figure 4.5: **Summary of the model for one neuron.** The plot displays the mean estimated contribution of images shown at each lag to the excitation of *neuron n64* \( r=0.59 \) at time \( t \). A two-phased sinusoid curve is fit to the observed values, and the lag corresponding to peak excitation (here \( t - 3 \)) is used for down-stream analysis. The frame rate is 60Hz. (Top) Direction and scale of impact for each feature category for *neuron n64*. Each feature in the dictionary was hand-assigned to a shape, a scale, a texture, and an orientation category (4 categories per feature). The impact of each category on the prediction depends on the magnitude and the sign homogeneity (consistently excitatory or consistently inhibitory) of the features belonging to the category. (Bottom) Most excitatory images according to the model. Both impact and the image selection are calculated based on the peak lag of the model.
4.2. RESULTS

An orientation invariant neuron selective for corners and acute curves:

In Figure 4.5B we display the most excitatory images ranked by predicted response, and the category impact values for neuron n64 (correlation coefficient r=0.67). The most excitatory images are quite similar and contain figures with straight angled and curved corners that are easily distinguished from their background. This is consistent with and highlighted by the bar-plot of impact values for feature categories. Both light and dark corners have the greatest positive impact. Other positive categories include straight edges, and dark features such as end-stops and curved edges. Textures and blobs do not excite this neuron, nor do specific orientations.

A neuron selective to orientation (V1-like):

Unlike n64, some neurons are selective for specific orientations. A prime example is neuron n71 (r=0.75) displayed in Figure 4.6A. This neuron is excited by images with strong elongated horizontal patterns, whether a single transition between dark and light or multiple bars. Both straight edges and bars have a strong positive impact, whereas scale and texture do not play a large role. The selectivity of this neuron resembles patterns found in V1 and V2 neurons.

A neuron selective for thin bars:

Neuron n40 presented in Figure 4.6B exhibits a third profile, distinct from the previous two. Thin bright bars on dark background excite this neuron. Other categories do not strongly impact this neuron, with the exception of acute bright features perhaps corresponding to bar ends. Though each thin-bar feature is oriented, there is no clear dominant orientation preference for this neuron. Note that n40 is the neuron best predicted by our model (r=0.77), and showed the greatest improvement in prediction compared to the Gabor model.

An orientation invariant texture neuron:

We find evidence of texture selectivity in many neurons. We present in Figure 4.6C neuron n59 (r=0.66), which predominantly responds to images with textured surfaces, such as flowerbeds, leaves, and mosaics. Consistently, features with high texture and multiple blobs have a strong positive impact on the firing rate. Straight edges do not excite this neuron. Several recorded neurons similarly prefer texture features to straight edges, and this distinction devises the main clusters in the neuron population examined in the next paragraph.
4.2. RESULTS

Figure 4.6: **Summary of three neuron models.** We plot three more summaries of the fitted models, for neurons with high prediction accuracy. Each neuron is represented by (top) a bar plot of category impacts, and (bottom) the most excitatory images for that neuron. Summaries for more neurons can be found in the Supplementary Figures.

A. Neuron n71 (r=0.75) is a V1-like neuron selective for horizontally oriented features.
B. Neuron n40 (r=0.77) is selective for thin bars.
C. Neuron n59 (r=0.66) is a texture neuron, selective for texture features and multiple bright patches.

### 4.2.5 V4 Neurons Cluster Into Two Groups: Those Selective to Image Contours Versus Those Selective to Texture

We use sparse principal component analysis or sparse PCA (Zou et al. 2006) to study the shape space spanned by the fitted neuron models. Sparse PCA reduces the dimension of the models by creating synthetic dimensions, each a combination of few categories. These combinations are optimized to conserve distances between the models, so that only neurons
with similar profiles will be close to each other in the reduced space.

Figure 4.7: Texture versus non-texture clusters. (The center panel displays the neuron-model population along the first and second sparse principle components. The size of a point reflects the r-score of the neuron, and the top 20% neurons are identified. The neurons cluster roughly into two main groups: those selective for texture (right) and those selective for non-texture. Texture neurons display a strong positive projection on the first component, and a weak projection on the second component, implying they do not have a horizontal/vertical preference. We visualize the high-scoring neurons found in both clusters by showing one excitatory image for each neuron (as in Figures 4.5 and 4.6).

Projected onto the largest two sparse principle components (sPCs), a coherent cluster of neurons is well separated from the others (Figure 4.7, center panel). We refer to these as texture neurons. These are neurons exhibiting a strong positive projection on the first component (x-axis), and weak projection on the second component (y-axis). All neurons in this cluster are all excited by strong textures and uneven surfaces, though variation in their preferences can be observed. 6 of the 13 highest scoring neurons (r>0.55) are among this cluster, and we depict those on the right panel, showing an excitatory image for each neuron. On the left panel we similarly display the high-scoring neurons that do not belong to the cluster.

To further interpret this cluster, we study the category combinations leading to the first two sPCs. Figure 4.8 displays (left column) the linear weights of each category combined to form the first five sPCs, and visualizes (right column) the sPCs in image space. The first sPC is excited by texture and shapes associated with texture, and is inhibited by straight edges. The second sPC, on the other hand, is excited by horizontal orientation and inhibited by vertical orientations. We conclude that neuron models exhibit a strong dichotomy: some neurons are excited by texture and do not have a strong orientation preference, whereas most neurons are excited by strong oriented edges.

Non-texture neurons are continuously distributed along other directions of variation, described by the next three sPCs (rows 3-5 of Figure 4.8). One dimension is the preference of scale, in
Figure 4.8: **Leading components of variation between neuron models.** Descriptions of the five leading linear contrasts explaining variation between neuron model summaries (k=19 category impact values per model), estimated using sparse PCA. On the left are the loadings of each component to the feature categories. On the right, a visualization of these contrasts in the stimulus space: the center-right images would respond negatively to the contrast, and far-right images respond positively. The first component differentiates texture features (positive) from edges and contours. The second component is a contrast of orientation: vertical with horizontal. A similar orientation contrast (diag 1 vs diag 2) can be seen in component 5. The final two shape components, 3 and 4, describe variation between neurons belonging to the larger non-texture cluster. Component 3 seems to separate by scale, whereas component 4 tells apart bars from corners and curves.

In particular the affinity to very narrow features such as bars and acute corners (see e.g. Figure 4.6B). This is represented by the third sPC. Another dimension, described by the fourth sPC, distinguishes preference of corners and curves over straight and oriented edges. The fifth sPC is similar to the second, distinguishing the two diagonal orientations.
4.3 Discussion

4.3.1 Relation With Previous Neurophysiological Studies Using Artificial Stimuli

Most previous studies on neuronal responses in the V4 area have used artificial stimuli such as gratings (Desimone and Schein 1987; Gallant et al. 1996) or simple shapes and objects (Kobatake and Tanaka 1994; Pasupathy and Connor 2002). In contrast, our data and model accommodate a rich set of features from natural images, and it is thus interesting to see whether our analysis confirms some conclusions drawn in these earlier studies. This is indeed the case concerning the diversity of V4 neurons, which respond to a wide array of stimuli characteristics such as curvature and orientation, and to some complex features containing multiple orientations. However, the most remarkable conclusion of our study is the prominent role of texture versus contour discrimination in V4. Other experimental works have suggested that V4 may be involved in this discrimination (Carlson et al. 2011; Desimone and Schein 1987; Gallant 2000; Gallant et al. 1993; Merigan 2000; Pasupathy and Connor 1999b), but all of these studies implicitly made the assumption that texture versus contour discrimination was important, whereas our data and model do not make such an explicit assumption and this discrimination role of V4 neurons falls out naturally from our data and model.

4.3.2 Relations With Previous Neuronal Models

Our approach produces neuron models that can accommodate for a richer set of image patterns compared to those used in previous studies, while modeling the invariance properties observed in V4 receptive fields. For example, David et al. (2006) modeled a neuron’s response as a linear combination of features obtained from the two-dimensional Fourier transform, each feature being selective for a spatial frequency and orientation in the receptive field. Our model includes such single orientation features, but also more complex ones that are selective for corners, curves, blobs, and other shapes. As a result, we obtained quantitative results (correlation between true and predicted neuron responses) that were significantly better than those reported by David et al. (2006). A closer agreement can be seen between our model and the principles set up by Cadieu et al. (2007): in their terminology, the sparse coding of features can be considered the “selectivity step”, and our pooling procedure their “invariance step”. The methodology they proposed, however, is limited by the incremental addition of Gabor inputs in the fitting stage, whereas ours uses learned complex features to efficiently represent natural images. In terms of prediction performance, the correlation scores reported by Cadieu et al. (2007) cannot be directly compared to ours since they used a simple stimulus set and images shown at 2Hz (versus 30Hz in our study).
4.3.3 Are the Receptive Fields Space-Time Separable?

Figure 4.5A shows a typical time response of a neuron. After an excitatory stimulus is shown to the subject, the neuronal activity typically attains a peak after a small lag, between 2 and 5 frames, before starting an inhibitory phase. It has been sometimes hypothesized that neuronal responses are space-time separable—that is, a neuron response should be the product of a time function and a spatial pattern function. This is in contrast to non-separable models, where responses to spatial patterns can arbitrarily change over time. For simple cells, it was found that the space-time separability assumption was approximately correct for some neurons but not all of them (DeAngelis et al. 1993; Mazer et al. 2002). We introduce in this chapter a new flexible technique for learning semi space-time separable models, which can interpolate between separable and non-separable models. This method, which is detailed in the methodological section, exploits low-rank representation of matrices (Fazel et al. 2001). It involves a regularization parameter chosen by cross-validation, which naturally interpolates between separable and non-separable regimes. It is therefore desirable to measure the amount of space-time separability in the models selected by cross-validation. This amount can be quantified by observing the rank of the learned parameter matrix (see methodological section), which can range from 1 (space-time separable) to 9 (non-separable). For 68 out 71 neurons, the ranks were spread between 3 and 6, which seems to confirm that interpolating between separable and non-separable models is an appropriate strategy for modeling V4 neurons.

4.3.4 Other Roles of V4: Color and Motion

Area V4 is known to play a role in the processing of color (Kotake et al. 2009) and was in fact originally believed to be mainly a color area (Zeki 1973). Motion is another important stimulus characteristic, which has been shown to be part of V4 neurons selectivity (Desimone and Schein 1987; Roe et al. 2012). We analyzed in our study neuronal responses to static monochrome images. Hence we did not investigate color and motion, but focused instead on the processing of shape and other spatial patterns. We believe that extending our model to handle naturalistic color movies is feasible and could help us understand other aspects of area V4. Orientation maps could for example be computed in the spatio-temporal domain, as already done for example with Gabor functions for analyzing the MT area (Nishimoto and Gallant 2011), and color could probably be handled by defining different color channels. One direction of our current research is to find the most biologically plausible and effective extension of our model, which should be both predictive and interpretable.
4.3.5 Biological Compatibility and Links With Computer Vision Models

We used an image representation inspired by state-of-the-art visual recognition techniques introduced in the computer vision community (Boureau et al. 2010; Lazebnik et al. 2006; Lowe 1999). A main drawback of such approaches is their lack of known links to biology or neurophysiology. Part of our efforts thus consisted in adapting ideas from these works to obtain invariant image representation, while keeping the model interpretable and as biologically compatible as possible. Building a fully biologically plausible model for V4 would require an understanding of the currently unknown functional organization of V4. Nevertheless, our model is mostly based on simple steps that are compatible with (or close to) our understanding of neuronal processing. For example, this is the case for our model’s first layer, which produces orientation maps by linear convolution of oriented filter and a nonlinear rectification step setting to zero the negative responses to the filter. This is exactly a standard model for V1 cells, which was originally designed based on biophysical considerations (Carandini et al. 1997; Movshon et al. 1978). The patches we extract from the orientation maps are then contrast normalized using a contrast response function similar to those observed in the monkey striate cortex by Albrecht and Hamilton (1982), see supplemental material. Interestingly, these orientation patches are also related to the concept of local image descriptor in computer vision such as the SIFT of Lowe (1999), which can be approximated by non-linear filtering steps similar to ours (see Bruna and Mallat (2012)).

Our model second layer is based on sparse coding principles, which were originally used in a neuroscience context (Olshausen and Field 1996). Previous studies have demonstrated the sparseness of the population response distribution to a given stimulus, as well as various nonlinear inhibition effects between neurons in V1 (Vinje and Gallant 2000). The third layer consists of a simple biologically compatible pooling operation related to energy models (Carandini et al. 1997). It is worthwhile to mention that the second and third layers follow recent “bags of visual words” techniques in computer vision (Boureau et al. 2010; Lazebnik et al. 2006), but use circular pooling regions better adapted to the shape of receptive fields.

4.4 Methods

4.4.1 Experimental Procedures

Extracellular neurophysiological recordings were made from 71 well-isolated neurons in area V4 from two awake, behaving, male rhesus macaques (Macaca mulatta). All procedures were performed under a protocol approved by the Animal Care and Use Committee at the University of California and met or exceeded National Institutes of Health and USDA
standards. Surgical procedures were conducted under appropriate anesthesia using standard sterile techniques (Vinje and Gallant 2002). We located area V4 by exterior cranial landmarks and/or by direct visualization of the lunate sulcus, and confirmed the location by comparing receptive field properties and response latencies to those reported previously (Gattass et al. 1988; Maunsell and Gibson 1992).

Experiments were controlled and stimuli generated using custom behavioral/stimulus display software (PyPE) running on a Linux-based PC. The stimuli were displayed on a 21” Trinitron monitor (Sony Inc.) capable of displaying luminances up to 500 Cd/m2. The monitor’s luminance nonlinearity (gamma) was calibrated and corrected in software to provide a linear luminance response. During recording, the animals performed a fixation task for a liquid reward. We monitored eye position with an infrared eye tracker (500 Hz: Eyelink II, SR Research, Toronto) and trials during which eye position deviated more than 0.5° from the fixation spot were excluded from analysis. The standard deviation of the fixational eye movements was typically 0.05°.

We recorded extracellular activity using tungsten electrodes (FHC, Bowdoinham, ME), and amplified, band-pass filtered, and isolated neural signals with a spike sorter (Plexon Inc., Dallas, TX). After isolating each neuron, we estimated the boundaries of the classical receptive field (CRF) using drifting bars and gratings. We then localized the CRF precisely by reverse correlation of responses to a dynamic sparse noise stimulus consisting of black and white squares or bars positioned randomly on a grey background (DeAngelis et al. 1993; Jones and Palmer 1987; Vinje and Gallant 2002). These squares were scaled so that six to eight squares spanned the manually estimated receptive field. The classical receptive field (CRF) was defined as the circle around the region where sparse noise stimulation elicited spiking responses. Our manual and automatic estimation procedures were generally in good agreement. Eccentricities ranged from 4° – 17° (median 6.3°) and RF diameters ranged from 5° – 15° (median 9.7°).

We probed neuronal behavior using natural image sequences. Each image was a circular patch cut from grayscale digital photographs from a commercial digital library (Corel Corp.). The photographs included landscapes, man-made objects, animals and humans. We chose patches using an automated algorithm that selected them at random, but favored patches with high contrast (to reduce the occurrence of blank stimuli, e.g. patches of sky). We concatenated images into sequences which were presented, centered on the CRF, at 2-4 times the diameter of the CRF, while the animal fixated. The display had a grey background matched to the mean luminance of the stimulus sequence. Since the original images were originally intended to be shown on a nonlinear CRT display, we preprocessed them with a gamma nonlinearity of 2.2, to give an appropriate luminance profile on the linearized monitor. The outer edges of the patches (10% of the radius) were blended linearly into the grey background of the display. Natural image sequences were updated at 30Hz.
Recordings were made for the 71 neurons by showing once sequences of 4,000-12,000 natural images at 30Hz (6028 images on average). The number of spikes for each neuron was measured at a 60Hz rate, resulting in two measurements per image. We then standardized the responses of each neuron to have zero-mean and unit standard deviation. This data was used to learn the parameters of our model. To measure the prediction accuracy, we also acquired test data, using a sequence of 300 images that were kept aside from the training set. The sequence of test images was shown several times (on average 9.3), and the responses were averaged to obtain a single response vector per neuron.

4.4.2 An Invariant Computational Model for V4 Tuned to Natural Images

V4 neurons have been shown to exhibit several types of invariance to visual stimuli, notably to the position of a stimulus within the receptive field. Our computational model for V4 (i) has invariant properties similar to those of V4 neurons, (ii) is interpretable and selective to different types of shapes and textures arising in natural images, (iii) significantly outperforms existing models in terms of prediction performance, and (iv) mostly relies on simple biologically compatible operations. We already provided a global description of the computational model in the results section and describe here the technical details involved in each model layer.

First layer - computation of orientation maps.

The 8 orientation maps produced by the first layer are obtained by keeping the positive response of the convolution between the input image and 8 small oriented filters of size 7 x 7 pixels. They are defined as first order derivatives of an isotropic two-dimensional Gaussian function with a one pixel standard deviation. In other words, the orientation maps are obtained by a linear filtering step followed by a simple non-linear rectification step. Before proceeding to the next layer, the orientation maps are spatially downsampled by a factor 8. This is performed by simple operations: convolution with a Gaussian filter with a standard deviation of 4 pixels (a step called anti-aliasing in the signal processing literature) and subsampling. Even though reducing the resolution of orientation maps might seem counter-intuitive since it discards some image information, it is a crucial step to achieve invariance to small shifts or deformations (which are only visible at a high resolution). Reducing the resolution in such a way can in fact be interpreted as an averaging or pooling procedure; it replaces the pixel values from 8 x 8 regions in the high-resolution orientation maps patches by a single pixel value. Since the input images in the dataset are of size 256 x 256, the (downsampled) orientation maps are of size 32 x 32.
Second layer contrast normalization and computation of feature maps.

The second layer goes beyond the computation of orientation maps. It makes our computational model not only selective to complex features that are tuned to natural images but also gain some invariance to local contrast changes. As mentioned earlier in this chapter, we leveraged the sparse coding formulation of Olshausen and Field (1996). Concretely, the output of the first layer can be seen as a three dimensional image, with two spatial dimensions and one dimension for orientation. The second layer extracts all overlapping three-dimensional patches of size 4 x 4 in the spatial domain and 8 orientations, which can be mathematically represented by real vectors of dimension 4 x 4 x 8 = 128. These three-dimensional patches are then contrast-normalized using a simple non-linear response function presented in the supplementary material. Using a dictionary of 2048 features that has been previously learned (we explain how to learn this dictionary in the next paragraph), each patch is encoded into a sparse vector of size 2048. Since there are patches at 29 x 29 different locations, the sparse coefficients are organized onto 2048 spatial feature maps of size 29 x 29.

How the dictionary is learned from data.

The procedure we have just described requires a dictionary of features. To build such an overcomplete basis set adapted to natural images, we randomly extracted two millions of three dimensional patches from orientation maps computed from our 4000 training images. We then used this set of patches in place of natural image patches in the formulation of (Olshausen and Field 1996), which involves the representation of every patch as a sparse linear combination of dictionary elements. We used the Matlab toolbox SPAMS (Mairal et al. 2010) that is dedicated to solving optimization problems such as the one arising in the sparse coding formulation of Olshausen and Field (1996). This so called “dictionary learning” procedure was performed offline, once for all. A dictionary size of 2048 elements is used, similarly as in computer vision approaches to object recognition and scene analysis (Boureau et al. 2010).

Third layer - spatial pooling of feature maps.

The output of the second layer consists of 2048 feature maps of size 29 x 29 resulting in a large number of parameters. The third layer significantly reduces this number with a non-linear pooling step for each feature map among 2048. Every feature map of size 29 x 29 is “summarized” by a few values, each one corresponding to the square root energy of a particular (predefined) “pooling regions. We considered in our experiments two different settings. The “full” model includes the 19 overlapping areas of different sizes that are presented in Figure 1. The “simplified” feature map includes only 5 non-overlapping pooling regions. The output of
4.4. METHODS

the third layer is therefore a high-dimensional vector of size \( p = 19 \times 2048 \) (full model) or \( 5 \times 2048 \) (simplified model). To alleviate any ambiguity in the hierarchical image processing description, all mathematical details are provided as supplemental material.

4.4.3 Prediction model - learning semi time-separable models with low-rank regularization.

The data collected consists of a sequence of neuronal responses \( Y_1, Y_2, Y_3 \), and encoded images \( X_1, X_2, X_3 \), where \( X_i \) is the invariant representation (obtained with the non-linear encoding pipeline presented in Figure 1 of the visual stimuli shown at time \( t \). We assumed that a response \( Y_t \) could be predicted from the previously seen images \( X_{(t-\tau)}, X_{(t-2)}, X_{(t-1)} \) as illustrated in Figure 2. This naturally led us to consider the following linear model

\[
Y_t \approx \sum_{i=1}^{\tau} W_i^T X_{t-i},
\]

where the vectors \( X_{t-i} \) are of dimension \( p \) and \( W = [W_1, W_2, \ldots, W_\tau] \) is a matrix of weights with \( p \) rows and \( \tau \) columns. The number of parameters in this prediction model is therefore quite large. For example, we used in our experiments \( \tau = 9 \) and the full model we considered has \( p = 19 \times 2048 \). Because the amount of training data is limited, we used a regularized learning procedure to avoid over-fitting. A standard regularization approach is to exploit a-priori knowledge to constrain the matrix of weights \( W \) we want to learn and reduce the number of parameters. One constraint stimulates that the model is fully time-separable—that is, the entry \( W_i(j) \) at column \( i \) and row \( j \) is the product of a time function \( Z(i) \) and feature function \( Y(j) \). Such a constraint reduces the number of parameters to learn from \( \tau \cdot p \) (the number of entries in \( W \)) to \( \tau + p \) (the number of entries in \( Z \) and \( Y \)). Unfortunately, time-separability leads to over-simplified models and we obtained relatively poor predictive performance using this constraint, motivating us to develop a more flexible approach. To do so, we noticed that time-separability in the weights \( W \) was mathematically equivalent to having the matrix \( W \) of rank one. A natural relaxation is thus low rank on a matrix \( W \) with low-rank. Fortunately, some regularization functions have been developed to impose the low-rank constraint, notably the so-called “trace norm” (Fazel et al. 2001). We considered a low-rank regularized least-square regression formulation using trace-norm and obtained what we call a “semi time-separable” prediction model. Learning such a model amounts to solving a convex optimization problem with a regularization parameter tuned with internal five-fold cross validation on the training set. As already demonstrated earlier in this chapter, such an approach leads to state-of-the-art predictive performance.
4.4.4 Inferring Neuron Selectivity From Fitted Prediction Model

Visualizing features and building a taxonomy.

Our model interpretation relies on the visualization and categorization of complex features learned with a sparse coding formulation. Unlike the original approach of Olshausen and Field (1996), the features are not learned on raw images but on 4 x 4 patches extracted from orientation maps, and can therefore not be directly displayed on a computer screen. Since these maps have been subsampled by a factor 8, any 4 x 4 patch from the orientation maps (let us call it $A$) represents a patch of size 32 x 32 in the original input image (we call it $B$). It is thus possible to indirectly “visualize” $A$ by displaying the original image patch $B$. Unfortunately such correspondence is not available for dictionary elements. We addressed this issue by building a database of two million matches between patches $A_1, A_2, A_3$, randomly sampled from the orientation maps of 4000 training images and their corresponding natural image patches $B_1, B_2, B_3, \ldots$. With this database in hand, we visualized a dictionary element $C$ by selecting the 50 patches from the database $A_1, A_2, A_3$, that are the most correlated with $C$. We then displayed their corresponding natural image patches from the set $B_1, B_2, B_3, \ldots$, as well as a mean representation of these 50 best matches. We show 12 examples of complex features in Figure 3(A) using this visualization tool and provide additional examples in the supplementary material. We remarked that the mean representation immediately allowed us to characterize the type of shape represented by a complex feature. However, the mean does not capture the variation of patches and can hide characteristics that are not aligned, such as texture. Displaying the best matching patches is thus useful for capturing these additional aspects. The next paragraph explains how to build a feature taxonomy using our visualization tool.

We manually annotated all basis elements from the learned overcomplete dictionary into categories that we defined based on empirical observations. More precisely, we defined four independent annotation schemes as follows:

- **Shape**: We associated to every feature one out of 8 possible shapes:
  - straight edges corresponding to white to black transitions;
  - bars, or patches with a single stripe;
  - multiple stripes;
  - curves and straight corners containing curved transitions between dark and bright areas including obtuse and right angles;
  - acute corners;
4.4. METHODS

- blobs (both bright and dark);
- complex blobs involving two white, two black or a complex combination of blobs (typically an indication of texture);
- other: features that do not fall into the previous categories, crosses, T-junctions, and more complex features that were hard to interpret (about 3% of the dictionary elements).

To illustrate this taxonomy, some of the categories are illustrated in Figure 3(B), where we display the mean representation of features with low texture content.

• orientation: we defined 5 orientation categories; a feature can be either non-oriented, or have a dominant orientation (horizontal, vertical and two diagonals). Even though we could have used a more refined annotation scheme (distinguishing left from right orientation for example), having simply 5 orientation categories made our fitted models easier to visualize and interpret.

• scale: we considered that the scale of an image feature represented by a dictionary element is small if it represents a shape that fits into a 16 x 16 natural image patch; medium if it fits inside the feature patch (32 x 32 pixels); large if the feature encodes part of a larger object.

• texture: we defined 3 categories to describe the interaction between a dictionary element and the texture content of a patch. Some dictionary elements have almost no texture content on top of the main shape, for example elements 4, 5, 6, 9 and 10 in Figure 3(A), while others have strong texture on top of the shape patches (elements 1, 3, 7, 8 11, and 12). An intermediate category, the texture border, classifies cases where the elements code a transition between textured and non textured areas (element 2).

Building this taxonomy requires hand annotating a single dictionary with 2048 elements, which can then be used to automatically analyze the 71 V4 neurons of this study.

Model summarization along taxonomy.

For the interpretation of a single neuronal model, we follow two complementary paths. First, we condense the large number of weights ($9 \times 5 \times 2048$) that define a model into a few ($k = 19$) impact values that measure the impact of each of the feature categories on the response, at the time of peak excitation. Second, we identify the set of images that, according to our model, evoked the greatest excitation for that neuron. We thus get both a quantitative summary of the model as well as a qualitative set of images from which the neuron function can be deduced.
4.4. METHODS

For both analyses, we focus only on the time frame (or rather the lag) corresponding to the peak excitation. Because different weights are used for each lag of the model, the same image impacts the response differently at different lags. For example, in almost all neurons (63/71) the most excitatory image would inhibit the neuron a few frames later. Furthermore, as a consequence of the low-rank regularization, the different lags of the model showed strong positive or negative correlations with each other (e.g. a median correlation of -0.54 between the most excitatory lag and the most inhibitory lag). We therefore restrict our analysis to the peak lag only. The curve of the averaged image contribution at each lag traces the temporal response function of a neuron. The peak is found by fitting a double-phased sinusoid to this curve and choosing the lag closest to the top of the curve. Only the weights reflecting this lag were analyzed downstream.

Impact values describe the effect on the response of a subset of features under a linear model. The impact of a feature category measures, in a single real value, two aspects of the features belonging to that category: (a) directionality do features from this category excite or inhibit the response and (b) magnitude to what extent do the features from this category change the response. Large impact values are given to categories whose features weights are both large and have a dominant directionality, either positive or negative. Impact values therefore extend the standard analysis of individual linear coefficients $D$ are they positive or negative, are they large or small $D$ onto sets of linear coefficients.

For a linear model, the contribution of feature $j(1jp)$ at time $t - l$ to the response at time $t$ is defined

$$\hat{y}_{t,l}^{(j)} = W_l(j)X_{t-l}(j),$$

where $X_{t-l}(j)$ marks the presence of the feature in the image shown, and $W_l(j)$ is the linear coefficient. For simplicity, because only one lag is considered we can drop the $l$ index. Then the feature has a positive (negative) contribution if the sign of $W(j)$ is positive (negative); furthermore, the magnitude of this contribution can be measured by the taking the standard deviation across different images, $SD_t(\hat{y}_t^{(j)})$. In a similar fashion, we take the contribution of a feature category $S$ to be the sum of the contributions of all features in the category, or $\hat{y}_t^{(S)} = \sum_{j \in S} \hat{y}_t^{(j)}$. Magnitude is measured by standard deviation across images as before, $m(S) = SD_t(\hat{y}_t^{(S)})$. Directionality is determined by the sign of the covariance between $\hat{y}_t^{(S)}$ and the contribution vector if all weights in the category were positive $\hat{y}_t^{(+S)} := \sum_{j \in S} |W(j)|X_t(j)$, hence $d(S) = \text{cor}(\hat{y}_t^{(S)}, \hat{y}_t^{(+S)})$. Finally, impact is defined to be the product of these two:

$$i(S) = m(S) \cdot d(S).$$

This definition of impact accounts for features being heterogeneous and for the patterns of correlation between features. Note, however, that the analysis of impact values is not very sensitive to alternative measures of magnitude and sign.

Impact values were computed for the four classifications schemes described above (8 shapes, 5 orientations, 3 scales and 3 textures) resulting in 19 values total. We present the impact
values computed only on the center region. Note however that the impact vectors computed on the center region are very similar to those computed on the other 4 pooling regions in almost all neurons (Corr>0.6 on 68/71 neurons). The resulting values and subsequent analysis using this summarization is not very sensitive to alternative measures of magnitude and sign.

**Analysis of the Neuron Population with Sparse Principal Component Analysis**

As input for this analysis we use the 19 category impact value vectors, centered and scaled, for each of the 71 neurons. We then compute the main directions of variance among the different neurons using sparse principal component analysis (Zou et al. 2006). The purpose of this analysis is twofold: (i) identify interpretable directions of variations; (ii) visualize clusters and continuums within the neuron population after projection along two or three principal components. The sparse PCA algorithm identifies a linear contrast that both captures a strong variation between the impact values of the different neurons and has only few non-zero values, sequentially repeating this process on the residual variation. 10 sparse directions of variance are identified using the ÒspcaÓ function of the Òelastic-netÓ package in R (regularization at 0.5), of which the first 5 directions account for 76% of the variance in the impact values.

**4.4.5 Estimation of explainable variance**

Explainable variance (EV) is the proportion of variance in the response that is not due to noise, and therefore can be predicted (explained) by a perfect deterministic model. EV estimates are used to calculate the percentage of explainable variance predicted (PP) by the model for each neuron, an accuracy measure that is less sensitive to the noise level differences between neurons.

The recorded data for each neuron in this experiment consist of many partially overlapping runs, which vary in length, so that each run covers only a partial stretch of the full validation sequence. Regardless of starting point and length of the run, images were always displayed according to the same sequence. The full test sequence Is composed of 300 images, with each image corresponding to two consecutive measurements. Let \( y^i_t \) denote the response measured for run \( i \) at time frame \( t \). Furthermore let \( \bar{y}^i_t \) be the average response across all runs that have measurements at frame \( t \), and \( \bar{y} \) the global average across frames.

This analysis follows the decomposition for conditional variance on the sequence \( \text{var}_t(y^i) = \text{var}_t(\mathbb{E}_i[y^i]) + \mathbb{E}_t[\text{var}_t(y^i)] \), corresponding to the total variance, the variance due to signal and the variance due to noise. To simplify notation we use \( \sigma_y^2 = \sigma_\mu^2 + \sigma_\epsilon^2 \) for the above equation.
The explainable variance is defined to be the ratio

\[
EV = \frac{\sigma_{\mu}^2}{\sigma_y^2} = \frac{\sigma_y^2 - \sigma_\varepsilon^2}{\sigma_y^2}.
\]

\(\sigma_y^2\) is estimated by using the sample variance across time \(\hat{\sigma}_y^2 = T(1 - 1) \sum_{t} (\bar{y}_t - \bar{y})^2\). Estimating \(\sigma_\varepsilon^2\) is more complicated, and in the following procedure relies on an assumption of independence of noise between runs. From the method of moments, the noise level (for a single run) at time \(t\) is estimated using

\[
s_\varepsilon^2(t) = (n(t) - 1)^{-1} \sum_i (y_{ti} - \bar{y}_t)^2,
\]

where \(n(t)\) is the number of runs that cover frame \(t\). \(s_\varepsilon^2(t)\) is expected to vary along the sequence, and may depend, for example, on \(\bar{y}_t\). The estimate for the variance at time \(t\) in the averaged sequence is \(\text{var}_\varepsilon(\bar{y}_t) = (s_\varepsilon^2(t))/n(t)\). Note that \(n(t)\) also often varies considerably for different locations in the sequence. For the overall noise level, the variance is averaged across the full sequence, in \(\hat{\sigma}_\varepsilon^2 = T^{-1} \sum_t \text{var}_\varepsilon(\bar{y}_t)\). The explainable variance is estimated by plugging in \(\hat{\sigma}_y^2\) and \(\hat{\sigma}_\varepsilon^2\) into Equation (*) . The proportion of predicted explainable variance is then \(r^2/EV\), where \(r\) is the correlation between the predictions and the average observed sequence.
Bibliography


Carandini, M., D. J. Heeger, and J. A. Movshon (1997, Nov). Linearity and normalization in
Cheung, M., T. A. Down, I. Latorre, and J. Ahringer (2011, June). Systematic bias in
high-throughput sequencing data and its correction by BEADS. *Nucleic Acids Research.*
M. Meyerson, and E. S. Lander (2008). High-resolution mapping of copy-number alterations
*Network* 16(2-3), 239–260.
mathematical modeling of neural systems.* The MIT Press.
DeAngelis, G. C., I. Ohzawa, and R. D. Freeman (1993, Apr). Spatiotemporal organization
of simple-cell receptive fields in the cat’s striate cortex. I. General characteristics and
short read data sets from high-throughput DNA sequencing. *Nucleic acids research* 36(16),
e105.
of the 2001*, Volume 6, pp. 4734–4739. IEEE.
Felsen, G., J. Touryan, F. Han, and Y. Dan (2005, Oct). Cortical sensitivity to visual features
Fox, M. D. and M. E. Raichle (2007). Spontaneous fluctuations in brain activity observed
with functional magnetic resonance imaging. *Nat Rev Neurosci* 8(9), 700–711.
Gallant, J. L., J. Braun, and D. C. Van Essen (1993, Jan). Selectivity for polar, hyperbolic,
Neural responses to polar, hyperbolic, and Cartesian gratings in area V4 of the macaque
BIBLIOGRAPHY


