

An Exploration of Kernels in Relation to the Applications of Bayesian Structured Sparsity

Pallavi Koppol

pkoppol@princeton.edu

Advisor: Prof. Barbara Engelhardt

bee@princeton.edu

Abstract

Over the past several decades, we have had the means to collect a rapidly increasing amount of data not only regarding the world around us, but also regarding ourselves. While this makes possible insights that were previously unattainable, one must also consider that much of the data we have collected may not actually be that informative. In this paper, we seek to draw insights in the field of genetics. We use the task of association mapping – identifying genetic variations that lead to the expression of particular traits – as an example of a scientific application where we have tremendous amounts of data (in the form of SNPs, commonly found genetic variations) and where only some of those data points actually give us meaningful information regarding results. We consider the idea of dealing with this challenge via a Bayesian approach to structured sparsity; this approach greatly depends on a Gram matrix of a Mercer kernel to dictate the similarity between data points. As such, we explore a variety of commonly used Mercer Kernels with the intention of understanding which, if any, is best suited to the task at hand. Furthermore, we strive to collect additional data in order to obtain more realistic Gram matrices and, ultimately, more realistic models and understanding of the mechanisms underlying phenotype expression.

1. Introduction

Our understanding of the world around us is informed by data. From the moment we are born until the moment we die, we are constantly observing data points and using them to draw conclusions regarding the world around us; this is the very nature of scientific exploration. It is true that the precise mechanisms by which humans store, process and act upon data is, perhaps, something of a mystery. Despite the general uncertainty surrounding the psychological and physiological mechanisms behind cognition of this sort, however, it is blatantly evident that in order for observations to be made about complex processes, one must have sufficient data points regarding the processes in question and must be able to synthesize that data in meaningful ways.

This understanding of the necessity of accumulating and synthesizing data has, in the past several years, taken society by storm. Indeed, we live in an era where *data* is a buzzword in virtually every field, and in which statistics and machine learning have become increasingly highly valued and sought after tools. These two observations are, of course, intrinsically linked and are undoubtedly byproducts of our increased societal capacity to save and store massive quantities of data. Virtually every functional member of society, for example, is in possession of a smartphone. These devices are able to monitor an individual's location at every hour of every day, and can even track things such as miles walked, flights climbed, blood pressure, etc. We have sleep-tracking devices, as well as applications that monitor how an individual is spending his or her free time.

Our data accumulation is not, however, limited to what one can do with their smartphone. We have the capacity now to store and share massive quantities of data regarding virtually every subject: genomics data, functional magnetic resonance imaging data, migration patterns of various animals, patterns of crime in cities, etc. Indeed, more and more of these large datasets are

being made publicly available, whether passively or through competitions, in an effort to entice people to make sense of these incredibly vast amounts of data.

This is to be expected – as the amount of data we are able to store increases, we are naturally in want of increasingly elegant ways in which to make sense of that data: as mentioned earlier, the data itself is fundamentally meaningless, but the conclusions that can be drawn from that data are incredibly desirous. Perhaps even more appealing than the conclusions that can be drawn, however, is the idea that those conclusions can guide us to take actions that allow us to maximize our own benefit – whether on an individual or a societal level.

In this paper, we seek to explore and understand novel ways of making sense of massive quantities of genetic information, which is a problem area which, we feel, has the potential to greatly affect society at large. We will begin by providing some background information regarding genetic information and the particular problem that we will be addressing.

Subsequently, we will discuss the statistical methodology that we will be considering. Following that, we will discuss some of the questions that we wanted to explore and provide a brief overview of how we went about finding answers to those questions. We will conclude by offering some thoughts and suggestions regarding future expansions upon this research.

2. Motivating Problem: Genetics

Genetics as a research area is particularly intriguing because it is applicable to everyone in society — furthermore, there are a myriad of diseases, disorders and other physiological phenomenon that are, as of yet, not fully understood. To make sense of the large amounts of genetics data that we have accumulated is to potentially start on the path to understanding some of these physiological abnormalities more, so that we may eventually

hope to go one step further and prescribe precautionary measures, or treatments to phenomenon that, currently, we can barely begin to fathom.

2.1. Single Nucleotide Polymorphisms

A single nucleotide polymorphism (which we will henceforth refer to as a SNP, as per the norm) is one type of naturally occurring genetic variation that can occur within individuals in a given population. To understand this further, a SNP can be understood to occur, as the name might suggest, when one nucleotide (adenine, cytosine, guanine, or thymine) is replaced by another in a given individual's genetic sequence [1].

The effects of these SNPs is an active and burgeoning area of research. The human genome contains several million SNPs, and while many of these SNPs are benign — in that they are not known to have effects on an individual's overall physiology, whether due to their location or other factors — other SNPs have the potential to serve as indicators of a myriad of diseases, disorders and various other physiological abnormalities and to help researchers identify disease-causing genes. This follows from the intuition that if a SNP occurs within, or in sufficient proximity to, a particular gene or regulatory region, it might modify the functionality of that corresponding gene. [1]

2.2. Association Mapping

In particular, we consider the process of association mapping. Association mapping refers to the task of "...identifying genetic variants that are associated with a quantitative trait, such as expression levels of a gene" [2]. Typically, the way this is done is by measuring the quantitative trait in question across n individuals and by using the SNPs as predictors. The ultimate goal of association mapping is to determine which SNPs have a non-zero correlation with the desired

quantitative trait — in other words, to be able to determine which SNPs are indicative of the trait in question (relevant to phenotype expression) [2].

2.3. HapMap Phase 3 Dataset

The primary dataset of interest with regards to this challenge is the HapMap Phase 3 dataset. This data is publically available for download, and has the overarching aim of developing a haplotype (set of genetic variations) map corresponding to the human genome. In particular, we used the same dataset that was used in [2]; that is, it consisted of 608 individuals and approximately 40 million SNPs per individual. Furthermore, in [2], 16,426 traits were used for the task of associative mapping. As in the [2] experiment, our intention was to simulate SNP data (i.e. establish a ground-truth detailing which SNPs did or did not have an effect on the final outcome), and compare our model's results to that simulated data. The efficacy of our model would then simply be the number of correct predictions made regarding a SNPs effect on the outcome.

3. Overview of Statistical Methodology

3.1. Sparsity

In understanding the statistical methods explored in this paper, it is important to first understand the concept of sparsity in general. Consider that a high dimensional feature set is one in which there are many features, but where the contribution of most of those features towards the outcome is negligible; in other words, most features are not indicative of what the outcome will be. Therefore, there will ultimately be a few features with non-zero contributions towards the outcome (hence the name *sparsity*); these features can

be thought of as the causal ones [3]. Sparsity is a particularly effective tool when the number of features exceeds the number of samples, when we want to increase tractability of parameter estimation, and when we want to simply understand which features are relevant to our problem and which are not (ie. we are less concerned with the effect size than whether or not an effect exists at all) [2].

3.2. Structured Sparsity

An initial approach to the issue of sparsity would be likely to assume some sparse, independent prior — this amounts to forcing most contribution weights to 0, unless the data presents strong evidence otherwise [2, 4]. In many scientific applications however, assuming this independent prior is not entirely accurate. Rather, we find that causal features tend to arise in groups; that is, causal features have dependence on each other [2,4]. The problem of association mapping is one of these scientific applications, as discussed earlier. Therefore, using a standard sparsity inducing model such as lasso, elastic net or ridge regression [12, 13, 20] would likely yield sub-optimal results. As a result, we look into methods that induce structured sparsity — i.e., models that maintain that only a few features contribute to the outcome, but that those features are dependent within themselves.

3.2.1. Bayesian Structured Sparsity

One approach to inducing structured sparsity is a Bayesian approach, as proposed in [2]. At a high level, this approach can be understood to operate via making use of a

gaussian process in order to share parameters across features as a function of their similarity. This approach is given in detail in [2], but we find it informative to briefly consider it in further detail, as the remainder of our research is directly affected by the nature of this Bayesian approach to structured sparsity.

In using this approach, we assume that there exist n samples and p predictors, such that we have a vector \mathbf{x} for each i th sample that consists of all of the predictor values for that sample, and a scalar y for each i th sample that represents the result. As stated in the paper, this can be encoded such that the predictors are given by $\mathbf{X} \in \mathbb{R}^{n \times p}$ and the results by $\mathbf{y} \in \mathbb{R}^n$. From this, Engelhardt and Adams show that the results are conditionally independent on predictors and several parameters, as shown:

$$\mathbf{y} | \mathbf{X}, \boldsymbol{\beta}, \beta_0, \nu \sim \mathcal{N}(\beta_0 \mathbf{1}_n + \mathbf{X}\boldsymbol{\beta}, \nu^{-1} \mathbb{I}_n).$$

For the sake of this paper, we are only concerned with the parameter $\boldsymbol{\beta}$, which represents the vector of weights; further information regarding the other parameters (as well as additional information regarding this approach) can be found in [2]. In this approach to structured sparsity, structure is induced into the distribution of the parameter $\boldsymbol{\beta}$ in such a way as to be dependent on Σ , which can be any given positive definite covariance matrix that is “used to specify the dependence structure for inclusion” [2].

4. Idea and Implementation

4.1. Kernels

Though in our chosen Bayesian approach to structured sparsity, as mentioned before, the covariance matrix Σ can be any given positive definite matrix, it is most compelling to explore the cases where it instead represents the pairwise similarity between predictors. This follows naturally from the idea that in a Gaussian process, the covariance matrix serves to define similarity [5]; by having a more accurate definition of similarity, one might expect to obtain more realistic results from their model — to relate this back to the problem of association mapping, it is intuitive to think that if we have a more realistic and highly detailed measure of similarity between SNPs, we will better be able to understand the mechanisms behind phenotype expression.

In particular, we are interested in exploring Σ such that it is a Gram matrix. A Gram matrix K is defined as having entries such that $K_{i,j} = k(x_i, x_j)$, where k is some kernel; we define a kernel to be some function that maps a pair of arguments $x, x' \in X$ into \mathbb{R} [5]. For example, in the case that k is a covariance function, then the Gram matrix K is necessarily equivalent to a covariance matrix – i.e., every element of K represents the covariance value between x_i and x_j [5, 6]. Furthermore, in that case, the Gram matrix K is also necessarily positive semidefinite [5].

With the presentation of the Bayesian approach to structured sparsity, reference was made to testing several kernel functions, but comparative results for those kernel functions in association mapping tests were not released. Therefore, we sought to explore and

compare the differing outcomes of several well-known kernel functions when used in conjunction with a Bayesian approach to structured sparsity.

4.1.1. Covariance Kernel Function

The first kernel function that we chose to implement was a straightforward covariance kernel function. That is [18],

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{(\mathbf{x}_i - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_j)}{n - 1}$$

This function was chosen primarily for its fundamental nature – several introductory papers on Gaussian processes and kernel functions make reference to it as a rudimentary kernel [5, 6, 7, 8], and we were interested in comparing its performance to the performance of peer kernels functions. This function was implemented using the numpy package for python, and by iterating through every combination of x_i and x_j (though, in truth, we only had to iterate through half of these combinations, as $k(x_i, x_j) = k(x_j, x_i)$).

4.1.2. Linear Kernel Function

We then implemented a linear kernel function. By definition [10],

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$$

This is actually a specific case of the polynomial kernel function, which we will further explore below. This kernel was selected for examination due to the fact that multiple studies regarding genetic variants opted to use this function (though not in the context of

structured sparsity) [7, 8]. We presume that this kernel would be most representative of data wherein the relationships between SNPs contains few interactions and is primarily linear. It is also worth noting that a linear kernel function might not capture dependencies or similarities between the individuals themselves. This kernel was implemented using the python packages numpy and scikit-learn, and by iterating over the array of predictors X in the same way described in 4.1.1.

4.1.3. Polynomial Kernel Function

The polynomial kernel function is, by definition [10]:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c_0)^d$$

where c_0 is some constant, and d is the function's degree. It is easy to see that when c_0 is equal to 0, and d is equal to 1, this is equivalent to the linear kernel that we discussed above. Truthfully, we are primarily interested in quadratic kernels, which were often used alongside linear kernels in studies pertaining to genetic variants[7, 8, 9]; unlike linear kernels, quadratic kernels seemed to perform better when interactions *did* exist between the SNPs and between the individuals themselves. Again, however, quadratic kernels were not used in the context of structured sparsity and especially not in the context of a Bayesian approach to structured sparsity. When implementing this function, we decided to generalize the quadratic kernel to a less specific polynomial kernel – this was with the anticipation that in future experiments, one might be interested in considering kernel functions of

varying degrees. Again, this was implemented using numpy and scikit-learn in the manner described in section 4.1.1.

4.1.4. Radial Basis Kernel Function

The radial basis kernel function is given by [10]

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

and gets its name from the fact that the function k is dependent only on $r = \|\mathbf{x} - \mathbf{x}'\|$ [5].

This function, as with the others we have discussed thus far, was attractive due to its appearance in studies regarding genetic variants and trait mapping. In particular, there was a study that showed that using a radial basis kernel function in conjunction with a support vector machine “achieved a 65.3% prediction rate” [11]. As with the other kernel functions, this function was implemented using a combination of the python packages numpy and scikit-learn, following the paradigm illustrated in section 4.1.1.

4.1.5. Mutual Information Kernel Function

The mutual information kernel is given by [14]

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{Q(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{Q(\mathbf{x}_i, \mathbf{x}_i)Q(\mathbf{x}_j, \mathbf{x}_j)}}$$

where the joint distribution

$$Q(\mathbf{x}_i, \mathbf{x}_j) = \int P_{med}(\theta)P(\mathbf{x}_i|\theta)P(\mathbf{x}_j|\theta)d\theta$$

given some mediator distribution $P_{med}(\theta)$. While this kernel was found less frequently in studies on genetic variants, it was mentioned in [2]. Though results regarding its

quantitative performance were excluded from [2], we felt it would be prudent to compare this kernel against several others in order to establish benchmarks. As with the remainder of the kernels, this kernel was implemented using the python packages numpy and scikit-learn, as per the method outlined in section 4.1.1.

4.1.6. Identical by State Kernel Function

The identical by state kernel function is of the form [15]

$$k(x_i, x_j) = \frac{\sum_{s=1}^p IBS(x_{i,s}, x_{j,s})}{2p}$$

where $IBS(x_{i,s}, x_{j,s})$ can take on the value of 0, 1 or 2 depending on how many alleles are the same between subjects x_i and x_j at SNP p . The IBS kernel function was of particular interest due to the fact that it is extremely domain specific, and was created with the task of trait-mapping in mind [15]. Furthermore, it was referenced alongside the linear and quadratic kernels in studies involving genetic variants. In studies it seems that, like the quadratic kernel, the IBS kernel performs better when moderate interactions exist between SNPs and between individuals [7]. In implementing this function, we took advantage of the fact that $IBS(x_{i,s}, x_{j,s})$ is essentially the manhattan distance between $x_{i,s}$ and $x_{j,s}$, and utilized the python packages numpy and scikit-learn, following the format described in section 4.1.1.

4.1.7. Kernel Construction

It is interesting and informative to note that “...the sums (and products) of valid covariance kernels give valid covariance functions (i.e. the resultant covariance matrices are positive (semi-)definite)” [6]. In the context of our exploration, this indicates that we may also consider any weighted linear combination of the aforementioned kernels as a viable kernel option to be used with a Bayesian approach to structured sparsity.

Furthermore, this property of kernel functions indicates that we can further incorporate additional information pertaining to the SNPs we have been considering without much hassle; indeed, we would simply need to find the kernel matrix produced for any additional information pertaining to the SNPs in consideration and linearly combine it with our original kernel matrix. We discuss the additional information in question in the following section.

4.2. Cis-Regulatory Elements Dataset

Cis-regulatory elements (CREs), as the name might suggest, regulate gene expression [16]. Furthermore, evidence exists that these CREs may be useful in understanding “how genetic transcription regulators, or eQTLs, replicate within and between cell types,” and in generally allowing us to further understand the mechanisms behind phenotype expression [16]. Because CREs affect the way genes are regulated, it is natural to wonder if CRE information will be able to shed any light on the association mapping task that we defined earlier — will SNPs located in areas of the genetic sequence close to, or even within, these regulatory elements share any significant properties? It would seem that if these CREs do provide additional, useful information regarding the relationship between SNPs, we would

be able to create more realistic and, therefore, more powerful kernel matrices than we would have been able to do otherwise.

In order to avoid problems caused by CRE cell-type specificity, we sought out datasets that came from lymphoblastoid cell lines. In particular, we used the following datasets:

1. *FAIRE-seq on GM12878*
2. *DNase-seq on GM12878*
3. *CTCF ChIP-seq on GM10248*
4. *E2F4 ChIP-seq on GM12878*
5. *IRF3 ChIP-seq on GM12878*
6. *CREB1 ChIP-seq on ECC-1*

These datasets were chosen carefully. (1) provides information regarding regions of open chromatin, and (2) provides information regarding the location of regulatory regions. Both are thought to be informative regarding SNPs [17, 22]. (3), (4), (5), and (6) identify the protein binding sites of CTCF, E2F4, IRF3 and CREB1 respectively — it seems intuitive to wonder if the location of a SNP with relation to a protein binding site will have an effect on phenotype expression. There seems to be evidence that variations in (6) play a role in susceptibility to bipolar disorder, indicating that this may be a CRE that does provide information regarding the relationship SNPs have between each other. [19]

In order to construct this dataset of CREs, the individual datasets were first downloaded from ENCODE (The Encyclopedia of DNA Elements) in .narrowPeak format. In this format, data is presented in the form: chromosome name, chromosome starting position, chromosome ending position, name, score (in a range from 0 - 1000, where the average value is between 100-1000), strand, signalValue, pValue, qValue and Peak [21]. We wrote a Python script that would run through each of these files and, for each different chromosome, would create a new file that contained two columns: the first column was base pair location (ie. 1 through the end of the chromosome, or the last position given in the downloaded dataset), and the second was either 0 or 1, depending on whether or not the score for that CRE was greater than 100 at that location. These files were then merged together such that for each chromosome, there existed one file containing CRE information for that chromosome.

5. Future Work and Expansions

The research conducted over the course of this past semester serves as a particularly good launching pad for several extensions, some of which are enumerated as follows.

5.1. Run Bayesian Structured Sparsity Code and Compare Kernel Efficacy

Ultimately, we seek to understand what kernel functions lead to the construction of a Bayesian model that is particularly well suited for inducing meaningful structured sparsity for the task of association mapping. Currently, while several frequently used and referenced kernel functions have been implemented, they have not yet been used in the construction of a Bayesian model for

structured sparsity; this is because, unfortunately, the code for the implementation of the method described in [2] is currently not completely function.

Therefore, the first order of business would be to ensure that this code run properly so that, as stated before, we will be able to establish a ground-truth determining which SNPs do or do not affect the final outcome and compare our model's results to that simulated data. The efficacy of the kernel and model could then be understood by simply finding the percentage of correct predictions made.

5.2. Explore Additional Kernels

Over the course of this semester, several well known and frequently used kernels, were implemented for use with a Bayesian approach to structured sparsity. A compelling and important area of further research would be to further implement a variety of other kernels in order to conduct more extensive testing to find the kernel best suited to this area of application. In particular, it might be useful to experiment with some very domain-specific kernels (ie. along the lines of the IBS kernel).

Furthermore, recall that over the course of this research, six cis-regulatory elements were collected into a dataset that we felt might provide further information regarding the relationship between any two given SNPs. In conducting more extensive testing, it would be valuable to consider creating a Gram matrix from the linear combination of weighted kernel functions (one kernel function would be run on the HapMap 3 data, and the other on

the corresponding cis-regulatory elements dataset) for use with the Bayesian approach to structured sparsity.

5.3. Expand Cis-Regulatory Elements Dataset

It is important to keep in mind that while our constructed dataset of six cis-regulatory elements may prove to be a reasonable starting point for more extensive and thorough tests, researching other cis-regulatory elements and incorporating them into this dataset will likely prove to be an important step in quantifying the relationship between SNPs: as with many things, the more data we have pertaining to each SNP, the better we will be able to model the relationship between them. Furthermore, being able to more effectively quantify the similarity between two SNPs will, by definition, result in more precise kernel matrices, which can then be used in correlation with the Bayesian Structured Sparsity approach we have discussed in order to better handle the challenge of association mapping.

5.4. fMRI Applications

Functional magnetic resonance imaging (fMRI) data seems to be well suited for use with the Bayesian Structured Sparsity approach to linear regression that we have been considering. The essence of capturing fMRI data is that subjects are presented with various stimuli, and the subjects' neuronal responses to those stimuli are recorded. This data is then processed into what are known as voxels, which are essentially cubes — ie. the data is represented in some 3-dimensional space — such that each voxel then contains some cluster of neurons. One could imagine this as being a 3-dimensional, grid like representation of the human brain. It would be an interesting challenge to try out various kernel functions on this type of data; because of the

nature of voxels (ie. clusters of neurons which will act together), this data is particularly well suited for a Bayesian approach to structured sparsity, as well. Furthermore, this is a well-motivated application, because by capturing these responses and trying to understand them, we can begin to understand and identify natural cognitive states. By extension, we can also start to identify dissociative or abnormal cognitive states.

5.5. Other Applications

In this paper, we have primarily discussed the relevance of Bayesian structured sparsity to the field of biogenetics, commonly used kernels within this field, and the methodology for the construction of more domain-specific kernels. We have also made reference to the applicability of this method to fMRI data., due to the nature of voxels. In future expansions of this research, it would be compelling to further identify and explore other fields to which this Bayesian structured sparsity approach might pertain and to explore the construction of kernels well suited to those domains. Another interesting possibility would be to test the kernels implemented here across various fields in order to determine which, if any, is the most viable ‘default’ option (in other words, to determine which of the kernels, if any, outperforms the others across data sets and fields).

6. Acknowledgements

First, I would like to thank Professor Engelhardt for providing me with the opportunity to explore an area that has proven to be both challenging and compelling, and for helping me understand the fundamentals of biogenetic research. This independent work research experience has been truly invaluable in that I have emerged with an understanding much better than that with which I entered (which was virtually none); for that, I feel truly fortunate.

Furthermore, I would like to thank Professor Norman and Professor Pillow of the Princeton Neuroscience Institute. Though, ultimately, my research diverged from fMRI applications, Professors Norman and Pillow were incredibly helpful and encouraging whenever I reached out to them.

I would also like to extend my sincerest gratitude towards my friends and family, without whom none of this would have been possible. They have been such a consistent source of encouragement, support and motivation over the course of the semester (and beyond!) and I am so lucky to be surrounded by such caring and uplifting people.

7. Honor Code

I pledge my honor that I have not violated the honor code during the writing of this paper.

/s/ Pallavi Koppol

8. Citations

[1] What are single nucleotide polymorphisms (SNPs)? Genetics Home Reference, 2015.

[2] B. Engelhardt and R. Adams. Bayesian structured sparsity from gaussian fields. arXiv preprint arXiv:1407.2235, 2014.

[3] M. Tipping. Sparse bayesian learning and the relevance vector machine. JMLR, 1:211–244, 2001.

[4] A. Wu, et. al. “Sparse Bayesian Structure Learning With Dependent Relevance Determination Prior”. In Advances In Neural Information Processing Systems (Nips). Montreal, Quebec, Canada, 2014. Print.

- [5] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, the MIT Press, 2006.
- [6] S. Roberts, M. Osborne, M. Ebdon, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Phil Trans R Soc A* 371: 20110550, 2013.
- [7] M. Wu, S. Lee, T. Cai, Y. Li, M. Boehnke and X. Lin, 'Rare-Variant Association Testing for Sequencing Data with the Sequence Kernel Association Test', *The American Journal of Human Genetics*, vol. 89, no. 1, pp. 82-93, 2011.
- [8] M. Wu, A. Maity, S. Lee, E. Simmons, Q. Harmon, X. Lin, S. Engel, J. Molldrem and P. Armistead, 'Kernel Machine SNP-Set Testing Under Multiple Candidate Kernels', *Genet. Epidemiol.*, vol. 37, no. 3, pp. 267-275, 2013.
- [9] H. Zhu, L. Li and H. Zhou, 'Nonlinear dimension reduction with Wright-Fisher kernel for genotype aggregation and association mapping', *Bioinformatics*, vol. 28, no. 18, pp. i375-i381, 2012.
- [10] Scikit-learn.org, '4.7. Pairwise metrics, Affinities and Kernels — scikit-learn 0.16.1 documentation', 2015. [Online]. Available: <http://scikit-learn.org/stable/modules/metrics.html>. [Accessed: 03- May- 2015].
- [11] H. Ban, J. Heo, K. Oh and K. Park, 'Identification of Type 2 Diabetes-associated combination of SNPs using Support Vector Machine', *BMC Genetics*, vol. 11, no. 1, p. 26, 2010.
- [12] R. Tibshirani. *Journal of the Royal Statistical Society. Series B*, pages 267–288, 1996.
- [13] H. Zou and T. Hastie. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

- [14] Matthias Seeger. Covariance kernels from Bayesian Generative Models. *Advances in Neural Information Processing Systems* 14, 2000.
- [15] L. Kwee, D. Liu, X. Lin, D. Ghosh and M. Epstein, 'A Powerful and Flexible Multilocus Association Test for Quantitative Traits', *The American Journal of Human Genetics*, vol. 82, no. 2, pp. 386-397, 2008.
- [16] C. Brown, L. Mangravite and B. Engelhardt, 'Integrative Modeling of eQTLs and Cis-Regulatory Elements Suggests Mechanisms Underlying Cell Type Specificity of eQTLs', *PLoS Genetics*, vol. 9, no. 8, p. e1003649, 2013.
- [17] Schaub, M. A., Boyle, A. P., Kundaje, A., Batzoglou, S., & Snyder, M. (2012). Linking disease associations with regulatory information in the human genome. *Genome Research*, 22(9), 1748–1759. doi:10.1101/gr.136127.111
- [18] GitHub, 'numpy/numpy', 2012. [Online]. Available: https://github.com/numpy/numpy/blob/v1.7.0/numpy/lib/function_base.py#L1947. [Accessed: 04- May- 2015].
- [19] M. Li, et.al., 'Allelic differences between Europeans and Chinese for CREB1 SNPs and their implications in gene expression regulation, hippocampal structure and function, and bipolar disorder susceptibility', *Molecular Psychiatry*, vol. 19, no. 4, pp. 452-461, 2013.
- [20] H. Zou and T. Hastie, 'Regularization and variable selection via the elastic net', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301-320, 2005.
- [21] Genome.ucsc.edu, 'UCSC Genome Bioinformatics: FAQ', 2015. [Online]. Available: <https://genome.ucsc.edu/FAQ/FAQformat.html>. [Accessed: 04- May- 2015].

[22] L. Song, Z. Zhang, L. Grassegger, A. Boyle, P. Giresi, B. Lee, N. Sheffield, S. Graf, M. Huss, D. Keefe, Z. Liu, D. London, R. McDaniell, Y. Shibata, K. Showers, J. Simon, T. Vales, T. Wang, D. Winter, Z. Zhang, N. Clarke, E. Birney, V. Iyer, G. Crawford, J. Lieb and T. Furey, 'Open chromatin defined by DNaseI and FAIRE identifies regulatory elements that shape cell-type identity', *Genome Research*, vol. 21, no. 10, pp. 1757-1767, 2011.