

A MAJOR QUALIFYING PROJECT REPORT
SUBMITTED TO THE FACULTY
OF THE
WORCESTER POLYTECHNIC INSTITUTE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF BACHELOR OF SCIENCE

BY

DONGNI ZHANG, MA

DATE: NOVEMBER 10TH, 2014

SPONSORED BY:
WORCESTER POLYTECHNIC INSTITUTE

PROJECT ADVISOR:
PROFESSOR ZHEYANG WU

Table of Contents

Chapter 1: Introduction	3
Chapter 2: Methodology.....	4
Research design	4
Seed gene detection	5
Candidate gene detection.....	5
Annotate nsSNVs according to protein binding sites.....	6
Likelihood ratio tests (LRT)	6
LRT for burden score weighted by binding site information (LRT-BS)	7
Chapter 3: Results.....	8
Simulation data analysis	8
Real data analysis.....	9
Chapter 4: Discussions	9
Chapter 5: Conclusions and Justification	9
References	10
Figures	11
Figure 1 - Gene detection working flow.....	11
Figure 2 - QQ-plot for CMC, C-alpha, LRT methods	12
Figure 3 - Systematic weighting experiment for simulated gene: SPTBN4, TPR and TCIRG1.	12
Tables.....	13
Table 1 - Seed gene list	13

Appendix.....	14
weightBurdenTest_invDir.R.....	14
weightBurdenTest_invDir_ToGetTypeIErrorCurve.R.....	18
Hypertension related gene list (based on literature).....	24
Hypertension centered network- interactions from HINT databse	24

Abstract

Statistical association studies have contributed significantly in the detection of novel genetic factors associated with complex diseases. However, there are various challenges in solving the missing heritability issue solely depending on statistical evidence of the association between genotype and phenotype, especially for sequencing data. Incorporation of biological information that reflects the complex mechanism of disease development is likely to increase the power of association studies for detecting novel disease genes. In this study, we develop a statistical framework for association studies that integrates the information of the functional effect of SNPs to the disease related protein-protein interactions. The method is applied to GAW19 exome sequencing data of uncorrelated individuals for detecting novel genes associated to hypotension. Based on both the real and simulated phenotypes of hypertension, the method is compared with multiple well-known association tests for sequencing data.

Acknowledgement

Without the help from Dr. Zheyang Wu, Dr. Dmitry Korkin, and Hongzhu Cui, the completion of this project would not have been possible. Thanks to their help of providing guidelines, sharing techniques and contributing to the results of this project. I would like to thank the Genetic Analysis Workshop for providing the Sequence, Blood Pressure and Expression Data.

Chapter 1: Introduction

Statistical association studies serve an important role in finding putative disease genes for further biological validation and reasoning. There are many association methods, but the common essential idea is to test the strength of statistical evidence for abnormal / non-random mutation distribution in cases and in controls. Such statistical evidence won't be strong or reliable if data sample size is relatively small and/or SNP mutations are rare, often the challenges imposed in current sequencing studies. Furthermore, the mechanism of genetic effect is complex, not a straight line from DNA to disease. A gene may be critical to a disease pathway, but the final disease status are affected by many other factors so that the association evidence measured strictly by genotype and phenotype data could also be weak. This is the challenge for finding more subtle disease genes to explain the missing heritability, especially after the low-hanging fruits have been picked. In order to address these challenges and increase the statistical power of association studies, it is promising to properly integrate biological prior information of the variants that reflect the middle steps of the genetic mechanism to disease development process.

In this study, we develop a statistical framework that allows incorporating prior information of SNPs into association tests. The basic idea is to prioritize SNPs that show prior importance to disease development. This is realized by relatively weighting the genotype of SNPs according to their prior information in testing a SNP set, which is treated as the functional unit of association, e.g., a gene. This framework has two major components. First, SNP weights are properly generated based on their functional annotation. Second, a likelihood ratio test (LRT) is constructed to incorporate these weights. To detect the presence of disease SNPs in a gene, LRT statistic is the ratio of the likelihoods between the null and the alternative regarding to whether the distribution in cases differ that in controls. Thus LRT is flexible to construct according to the meanings of the null of no association and the alternative while incorporating informative weights. Furthermore, LRT is optimal for detecting weak and sparse signals {Ingster, 1997 #2436;Yang, 2014 #2444}. There are different versions of

LRT, but we adapt a formulation by Chen et al. {Chen, 2013 #2486} as the prototype statistic for sequencing data.

Protein-protein interactions (PPI) are one important component related to disease development. Disease gene may function through influencing PPIs. Several recent genome-wide association studies (GWAS) have reported the value of incorporating PPI information into the pipeline of identifying novel genes of Type 1 diabetes and kidney dysfunction {Bergholdt, 2012 #2484;Chasman, 2012 #2487}. However, their methodology mainly uses generic functional information, *e.g.*, GO terms, to filter candidate genes and SNPs for test, but the association test itself is traditional without incorporating such information {Consortium, 2007 #2909}. Filtering will loosen the strict genome-wide significance level in favour of relatively weak association signals of true functional SNPs and genes, but it would be nicer to drop as few genes as possible, and combine the prior biological information with association test process in a quantitative fashion. In this study, we predict and annotate the effect of SNPs to disease related PPIs. Then we convert the annotation into a component of SNP weighting scheme for incorporating into the LRT test. The purpose is not to restrict the gene candidates but to use PPI information to improve the ranking profile of all genes.

Chapter 2: Methodology

Research design

To implicitly incorporate the PPI information for aiding gene hunting, we estimate and employ the weighting scheme of variants according to their effects on PPIs and study their group association by using a flexible LRT testing procedure. Figure 1 shows the workflow of four steps to achieve the goal. First, we determine the seed genes associated with Hypertension based on literature, data evidence and other related sources (Welter, D. et al, 2014; Online Mendelian Inheritance in Man, OMIM). Second, we analyse PPI network (physical interactions

only) centering on these seed genes to find out candidate genes associated with Hypertension. The candidate genes will be utilized to study the statistical association with Hypertension (Szklarczyk, Damian, et al, 2011). Third, we create annotations for SNPs regarding their impact on the several aspects of PPI networks. Fourth, we incorporate the SNP annotation information (weighting score) into the statistical association study by the LRT test. In the following, we give details of each step.

Seed gene detection

The “seed genes” are obtained from OMIM (Online Mendelian Inheritance in Man) by 1) Searching the key word: hypertension OR (“high blood pressure”), and 2) Extracting only genes with direct “phenotype- gene relationships”. The original gene list from step (1) yielded 290 genes. Step (2) is performed by manually justifying each gene and taking only the entries with “phenotype-gene relationship” recorded on OMIM. See result section for a list of 21 genes that was obtained as the final seed genes.

Candidate gene detection

The “candidate genes” are obtained from multiple sources. First, HINT (High-quality INTERactomes) from Yulab, is a database containing high-quality PPIs, which are compiled from various sources. It takes gene(s) as input and output an interactive image of its PPI network. We input seed genes into HINT, select interaction type “Binary”/ “co-complex” respectively, and enable the high-quality filter to obtain only high-quality interactomes. The genes in the PPI network are considered as candidate genes. [See the figure for seed genes and their neighbourhood genes.]

Second, Human Interactome Database (<http://interactome.dfci.harvard.edu/index.php?page=home>) is part of the CCSB project, which takes efforts to map the human binary interactome. Their long-term goal is to generate and analyze high-quality yeast two-hybrid (Y2H) interactions at high-throughput for all pairwise combinations of predicted gene products for which there is at least one Gateway-cloned ORF available (Human

ORFeome website). All individual datasets, including our most recent unpublished data, are described below with hyperlinks to the prepublication data or the relevant publications.

Third, **the association Test $C(\alpha)$** enables the test for rare variants disease association, under the assumption that the rare variants in cases and controls are a mix of deleterious, protective and neutral variants. The $C(\alpha)$ statistic is computed as follows:

$$T = \sum_{i=1}^m [(y_i - n_i p_0)^2 - n_i p_0 (1 - p_0)]$$

The $C(\alpha)$ test would yield a list of variants with considerably low p-values, which could be considered as candidate genes. The cut-off of p-value for $C(\alpha)$ test is set to 0.001.

Fourth, the association test CMC (Combined and Multivariate Collapsing method for rare variants) tests for rare variants disease association as well. CMC method groups variants by gene. The CMC method uses Fisher's test statistic to avoid the computationally intensive permutation procedure. The input would be our raw data. Output variants with low p-values would be considered as candidate genes. The cut-off of p-value for CMC test is set to 0.002.

Annotate nsSNVs according to protein binding sites

We annotated non-synonymous SNVs by sequence-based prediction on whether they locate in protein binding sites. In particular, based on the variant information provided in the VCF files of the odd numbers of chromosomes for uncorrelated individuals, applying ANNOVAR [citation], we annotated 4,457 nsSNVs on 2,711 genes regarding to whether they are located on protein binding sites or not.

Likelihood ratio tests (LRT)

For a gene g_j (or a functional group of SNP set), the generic LRT formula is

$$\Lambda(g_j) = \log(L^A L^U / L), \quad (1)$$

where L^A , L^U , and L are the likelihoods for the distributions of an appropriate disease-association measure in cases, in controls, and in both groups. The numerator of LRT separates the likelihoods in cases and in controls to model the alternative hypothesis that there exists an association in terms of the differentiation between the two groups; the denominator pools the data of cases and controls together for the likelihood of the null hypothesis of no association. Here we adapt an LRT based on Bernoulli likelihoods {Chen, 2013 #2486}:

$$\Lambda(g_j) = \log \frac{(\hat{p}_j^A)^{T_j^A} (1-\hat{p}_j^A)^{(m-T_j^A)} (\hat{p}_j^U)^{T_j^U} (1-\hat{p}_j^U)^{(m-T_j^U)}}{(\hat{p}_j)^{T_j} (1-\hat{p}_j)^{(m+l-T_j)}}, \quad (2)$$

where m and l are the number of cases and controls, T_j^A , T_j^U and T_j (\hat{p}_j^A , \hat{p}_j^U and \hat{p}_j) are the total numbers (and the corresponding estimated proportions) of the burden scores that exceed a threshold t in cases, controls and both groups, respectively.

The burden scores are the collapsed genotypes over SNVs on a gene, which measure the overall mutation distributions {Morgenthaler, 2007 #2847}. Specifically, for the j th gene of the individual k , the burden score is

$$S_{jk} = \sum_{i=1}^{n_j} x_{ik}, \quad (3)$$

where x_{ik} is the genotype of SNP i of individual k . We search a sequence of threshold t and choose the value that maximizes the test statistic. When $\hat{p}_j^A \leq \hat{p}_j^U$, the test statistic is adjusted to be

$$\Lambda(g_j) = \log \frac{(\hat{p}_j^U)^{T_j^A} (1-\hat{p}_j^U)^{(m-T_j^A)} (\hat{p}_j^A)^{T_j^U} (1-\hat{p}_j^A)^{(m-T_j^U)}}{(\hat{p}_j)^{T_j} (1-\hat{p}_j)^{(m+l-T_j)}}. \quad (4)$$

We apply permutation-based test to calculate p-values, which accommodates the departure of assumptions not necessarily satisfied in real data to control the type I error rate. We implemented the LRT and permutation test by R functions, and applied Variant Tools {San Lucas, 2012 #2441; Wang, 2014 #2440} to manipulate data and to call these R functions.

LRT for burden score weighted by binding site information (LRT-BS)

This modified LRT method similar as above but with a weighted burden score

$$S_{jk} = \sum_{i=1}^{n_j} s_i x_{ik}, \quad (5)$$

where s_i is the weight to indicate the importance of variants according to whether they are located on protein binding sites.

Chapter 3: Results

Simulation data analysis

GAW19 exome sequencing variants on odd numbers of chromosomes for uncorrelated individuals are used as the genotype data. Phenotypes are the hypertension status (about 336 cases and 1,607 controls) of the 200 simulations. Using the simulated data, we can study the statistical power and type I error rate control of the association tests.

First, we study the strategy to weight the functional SNVs. Figure 1 shows the power curves of three genes containing simulated functional SNVs: SPTBN4 (functional SNVs have relatively strong beneficial effects), TPR (functional SNVs have relatively weak effects), and TCIRG1 (functional SNVs have relatively strong deleterious effects). We arbitrarily select 10 functional SNVs and assign weight (2, 5 or 10), the rest functional SNVs are pretended unknown, and they and the rest irrelevant SNVs are assigned weight (1 or -1). Figure 1 shows that different weighting schemes perform differently, but in general the combination 1-10 is relatively good and robust. Thus, we assign potentially functional SNVs and the rest weights that have the same sign but with relatively big difference. For the nsSNVs in binding site, we assign weight 10; for nsSNVs on in binding site, we assign weight 5; and for the synonymous SNVs, we assign weight 1.

Second, we compare LRT and the LRT-BS regarding to their statistical power and size.

Real data analysis

The phenotype data is based on the real DBP and SBP, for which hypertension cases are defined as individuals with $SBP \geq 140$ or $DBP \geq 90$ according to the American Heart Association. Thus for the real data we have 394 cases and 1,457 controls.

Four methods are applied to the association tests for genes (RefGene): C- α test {Neale, 2011 #2850}, CMC test {Li, 2008 #2822}, LRT, and LRT-BS. Table 1 shows the top genes (with p-values $< 2.0E-4$) yielded by C- α test (4 genes), CMC (3), LRT (5), LRT-BS (8). At the same level of the genomic inflation factor λ , LRT-BS has three more genes discovered than LRT, which include gene MSL2, a gene also discovered by CMC with a good inflation level of $\lambda=1.03$. Thus LRT-BS is more powerful than LRT.

Chapter 4: Discussions

One of the limits of current method is that genomic inflation factor is relatively big, which needs further research and modification.

The annotations for whether a SNV is on the binding site has limit in specifying the influence of the SNVs to the protein-protein interactions. Further annotations on the SNV effect (beneficial, deleterious, or neural) to individual PPIs and the PPI network as a whole will provide more information. LRT has the potential to incorporate different levels of the information by constructing multiple components of the test statistics. We will test on this idea later as long as we get the improved estimation of the SNV effects.

Chapter 5: Conclusions and Justification

Biological information on SNVs can be used to improve the association tests. We incorporate the information on whether nsSNVs are located in protein binding site into a LRT test, which has the similar genomic inflation

factor as the original LRT test but provide more power to detect more significant genes. Among the putative genes discovered by multiple methods, MSL2 are discovered by both CMC and LRT-BS, and is worth further validation study.

References

Online Mendelian Inheritance in Man, OMIM[®]. McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD), {Ayers, #2703}. World Wide Web URL: <http://omim.org/>

Price, A.L., Kryukov, G.V., de Bakker, P.I., Purcell, S.M., Staples, J., Wei, L.J., and Sunyaev, S.R. (2010). Pooled association tests for rare variants in exon-resequencing studies. *Am. J. Hum. Genet.* 86, 832–838.

Šali A, Potterton L, Yuan F, van Vlijmen H, Karplus M: Evaluation of comparative protein modeling by MODELLER. *Proteins: Structure, Function, and Bioinformatics* 1995, 23(3):318-326.

Šikić M, Tomić S, Vlahoviček K: Prediction of protein–protein interaction sites in sequences and 3D structures by random forests. *PLoS computational biology* 2009, 5(1):e1000278.

Szklarczyk, Damian, et al. "The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored." *Nucleic acids research* 39.suppl 1 (2011): D561-D568.

Understanding Blood Pressure Readings. American Heart Association. April 4th, 2012.
http://www.heart.org/HEARTORG/Conditions/HighBloodPressure/AboutHighBloodPressure/Understanding-Blood-Pressure-Readings_UCM_301764_Article.jsp

Welter, D., MacArthur, J., Morales, J., Burdett, T., Hall, P., Junkins, H., ... & Parkinson, H. (2014). The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic acids research*, 42(D1), D1001-D1006.

Figures

Figure 1 - Gene detection working flow.

Four steps are involved in gene detection. Two main components: Estimation of SNP annotations on their effects to PPI; Association test by LRT.

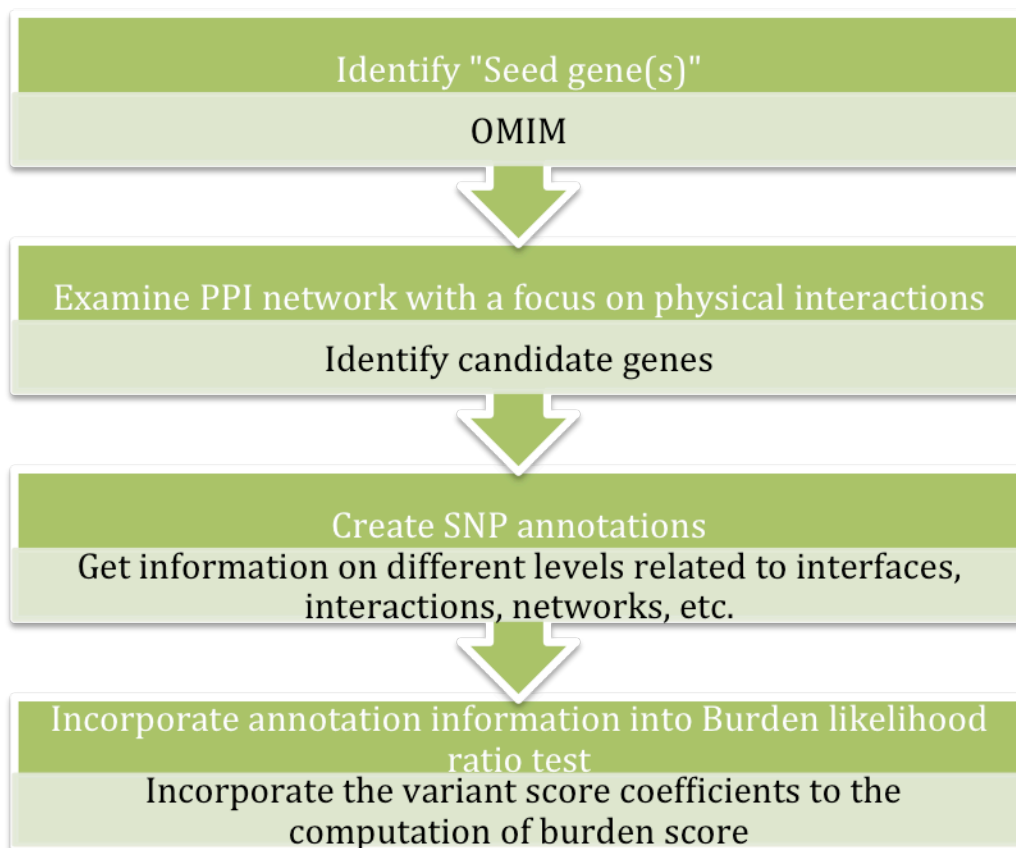


Figure 2 - QQ-plot for CMC, C-alpha, LRT methods

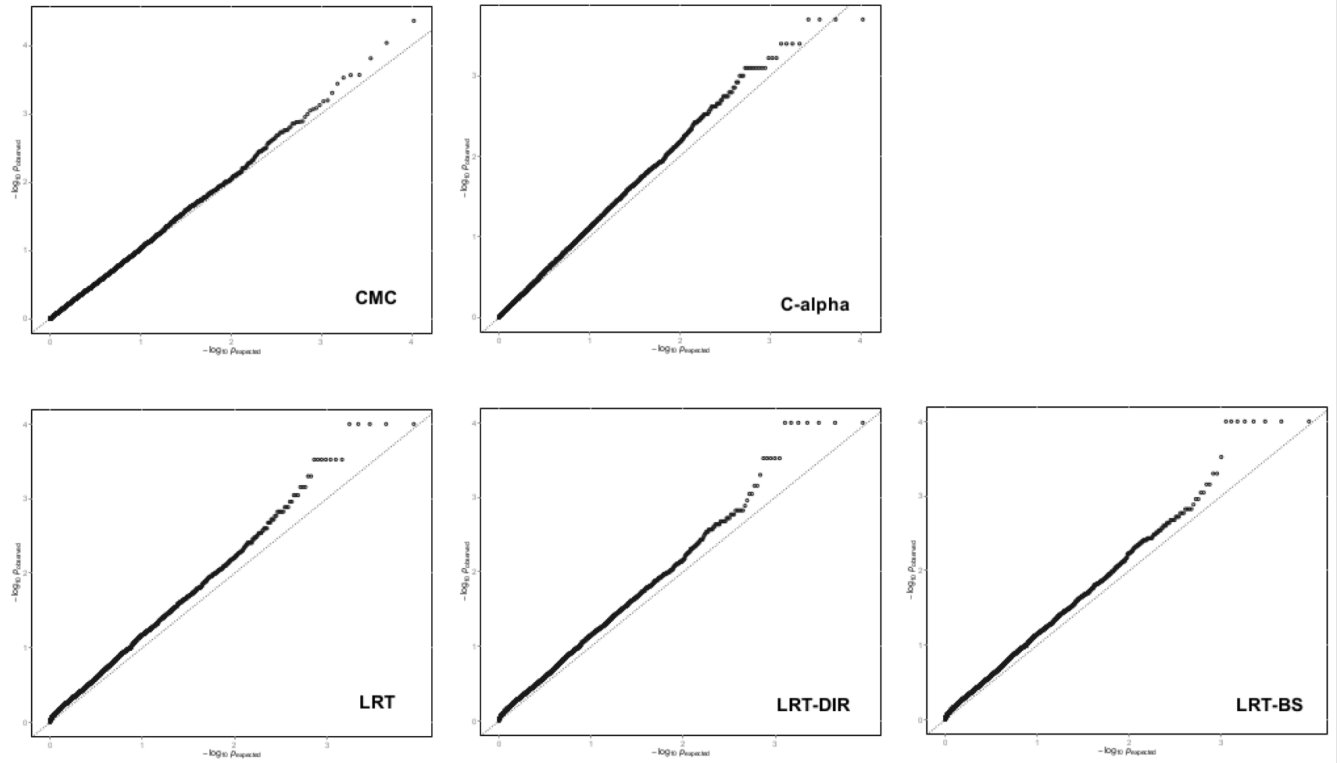
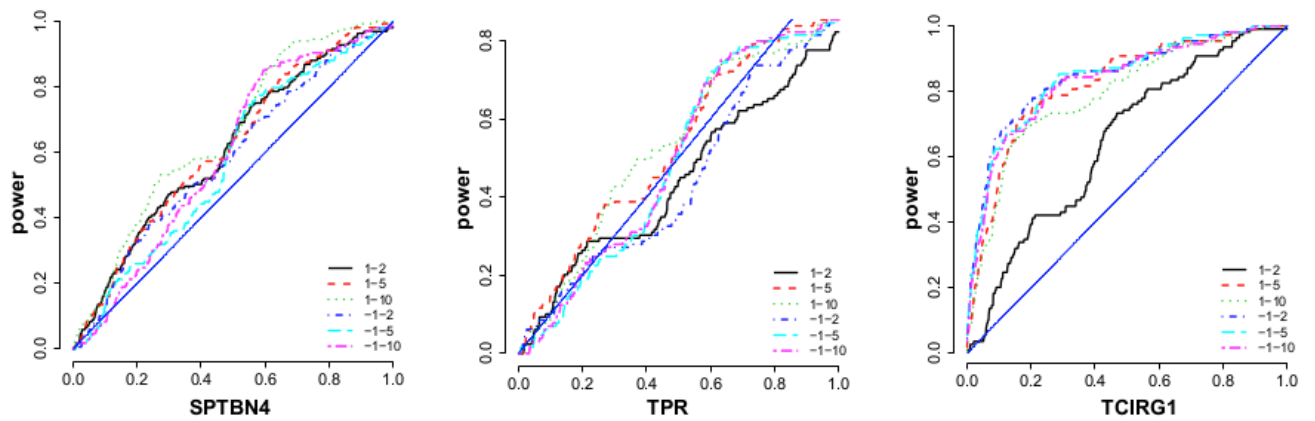


Figure 3 - Systematic weighting experiment for simulated gene: SPTBN4, TPR and TCIRG1.



Tables

Table 1 - Seed gene list

Top genes and their p-values by four association tests (λ is the genomic inflation factor).

Genes	Chr	C- α ($\lambda=1.11$)	CMC ($\lambda=1.03$)	LRT ($\lambda=1.27$)	LRT-BS ($\lambda=1.28$)
C7orf55- LUC7L2	7	1.99E-04	-	-	-
LUC7L2	7	1.99E-04	-	-	-
DNAH9	17	1.99E-04	-	-	-
AKAP8	19	1.99E-04	-	-	-
MSL2	3	-	4.39E-05	-	< 1.00E-4
ZBTB4	17	-	9.21E-05	-	-
BCHE	3	-	1.55E-04	-	-
LZIC	1	-	-	< 1.00E-4	< 1.00E-4
ABO	9	-	-	-	-
COL15A1	9	-	-	< 1.00E-4	< 1.00E-4
PSPC1	13	-	-	-	< 1.00E-4
TLN2	15	-	-	-	< 1.00E-4
ZNF557	19	-	-	-	< 1.00E-4
ITGA2	5	-	-	< 1.00E-4	-
ZMYM5	13	-	-	-	< 1.00E-4

YIPF2	19	-	-	< 1.00E-4	< 1.00E-4
PAEP	9	-	-	< 1.00E-4	-

Appendix

weightBurdenTest_invDir.R

```
# BEGINCONF
# [geneSize]
# [permuN]
# [pvalue]
# ENDCONF
```

#Note this function give the inversed direction weight, contrasting to weightBurdenTest.R

```
#####
```

#Description: Interface R function to be called by Variant Associate Tools. It has the choice to calculate an association test statistic based on direction weighted burden score. That is, if MAF in cases < MAF in controls, direction weight = 1, o.w., =-1. Further explicit weights can be multiplied by these direction weights to get the final weights. Permutation test is used to get p-values. Direction weights are re-calculated after each permutation.

#Depends on:

#Arguments: dat: data from Variant Tools. Require: (a) For phenotype, 1s indicate cases; 0s indicate controls. (b) genotype scores are counts of the minor alleles (mutations).

```
# Yphenotype.name: Name of the phenotype
```

```
# testFunc: The name of the association test statistic calculating function
```

```
# weightScheme: "noWei": no weighting; "expWei": explicit weight only; "dirWei": direction weight only; "expDirWei": explicit weights are multiplied b direction weights to get the final weights. Default is no weighting. Variant Tools command must specify the expliciate weight for "expWei" and "expDirWei";
```

```
# permuN: Number of the permutations
```

#Details:

#Value: Size of snp group; # of permutations; pvalue

#Note:

#Author(s): ZWu, DZhang

#References:

#See Also:

#Example:

```
#####
```

```

weightBurdenTest_invDir = function (dat, phenotype.name = "HTN", testFunc="LRTStat",
weightScheme="noWei", permuN = 5000) {

  phenotypes = dat@Y[,phenotype.name]; #phenotype
  caseN = sum(phenotypes==1);
  controlN = sum(phenotypes==0);

  genotypes = dat@X; #Genotype
  genotypes[is.na(genotypes)] = 0; #Missing genotype are treated as the major allele.
  #genotypes= round(genotypes);

  #calculate direction weights based on genotypes and phenotypes
  if (weightScheme %in% c("noWei", "dirWei")) weights = data.frame(rep(1, ncol(genotypes)));
#no weighting
  if (weightScheme %in% c("expWei", "expDirWei")) weights = dat@V;
  #explicit weights
  if (weightScheme %in% c("dirWei", "expDirWei")) {
    DirectionWeight = unlist(apply(genotypes, 2, function(g) {
      maf.case = sum(g[phenotypes==1])/(2*caseN);
      maf.contr = sum(g[phenotypes==0])/(2*controlN);
      #ifelse(maf.case >= maf.contr, 1, -1); #set up in weightBurdenTest.R
      ifelse(maf.case >= maf.contr, -1, 1);
    }));
    weights = weights*DirectionWeight;
  }

  #For real data
  stat = do.call(testFunc, list(phenotypes, genotypes, weights, caseN, controlN));

  #For permutation test
  stat.perms = 0;
  for (i in 1:(permuN/10)) {
    pheno.perm = sample(phenotypes);
    #Permutation stage 1
    if (weightScheme %in% c("noWei", "dirWei")) weights = data.frame(rep(1,
ncol(genotypes)));
    if (weightScheme %in% c("expWei", "expDirWei")) weights = dat@V;

    if (weightScheme %in% c("dirWei", "expDirWei")) {
      DirectionWeight = unlist(apply(genotypes, 2, function(g) {
        maf.case = sum(g[pheno.perm==1])/(2*caseN);
        maf.contr = sum(g[pheno.perm==0])/(2*controlN);
        #ifelse(maf.case >= maf.contr, 1, -1); #set up in weightBurdenTest.R
        ifelse(maf.case >= maf.contr, -1, 1);
      }));
      weights = weights*DirectionWeight;
    }
    stat.perms[i] = do.call(testFunc, list(pheno.perm, genotypes, weights, caseN, controlN));
  }
}

```

```

}
pValue = (sum(stat.perms >= stat)+0.5)/(length(stat.perms)+1);
if (pValue > 0.1) {
  return (list(geneSize=ncol(genotypes), permuN=i, pvalue=pValue));
}else {
  for (i in (permuN/10+1):permuN) {
    #Permutation stage 2
    pheno.perm = sample(phenotypes);
    if (weightScheme %in% c("noWei", "dirWei"))      weights = data.frame(rep(1,
ncol(genotypes)));
    if (weightScheme %in% c("expWei", "expDirWei"))  weights = dat@V;

    if (weightScheme %in% c("dirWei", "expDirWei")) {
      DirectionWeight = unlist(apply(genotypes, 2, function(g) {
        maf.case = sum(g[pheno.perm==1])/(2*caseN);
        maf.contr = sum(g[pheno.perm==0])/(2*controlN);
        #ifelse(maf.case >= maf.contr, 1, -1); #set up in weightBurdenTest.R
        ifelse(maf.case >= maf.contr, -1, 1);
      }));
      weights = weights*DirectionWeight;
    }
    stat.perms[i] = do.call(testFunc, list(pheno.perm, genotypes, weights, caseN,
controlN));
  }
  pValue = (sum(stat.perms >= stat)+0.5)/(length(stat.perms)+1);
  return (list(geneSize=ncol(genotypes), permuN=i, pvalue=pValue));
}
}
}

```

```

#####
#Description: Calculate the LRT statistic according to Bernoulli distribution. The burden score can be
weighted. The LRT statistic is gotten at the best burden score threshold that maximizes the statistic.
#Depends on:
#Arguments: phenotypes: A vector of binary phenotype values. Require: 1s indicate cases; 0s indicate
controls.
#          genotypes: Genotype data frame (cols: variants; rows: sample subjects)
#          weights: A one-col data frame vector of variant weights wrt to genotypes. Default: all
1s (no weighting).
#          caseN: Number of cases;
#          controlN: Number of controls.
#Details:
#Value: LRT statistic
#Note:
#Author(s): ZWu, DZhang

```

```

#References: Chen, ..., Karchin 2013 PloS Paper
#See Also:
#Example:
#####
LRTStat <- function(phenotypes, genotypes, weights=data.frame(rep(1, ncol(genotypes))), caseN,
controlN)
{
    burdenScores = data.matrix(genotypes) %*% data.matrix(weights);

    thresholds = c(unique(burdenScores), max(burdenScores)+1);
    stats = array(0, dim=c(length(thresholds) ));

    for (i in 1:length(thresholds) )
    {
        Ys = (burdenScores >= thresholds[i]);
        Ycase = sum(Ys[phenotypes==1]);
        Ycontrol = sum(Ys[phenotypes==0]);
        stats[i] = LRTStat.condi(Ycase, Ycontrol, caseN, controlN);
    }
    finalStat = max(stats[!is.na(stats)]);
    return(finalStat);
}

```

```

#####
#Description: Calculate the LRT statistic, conditional on given numbers of 1's in cases and controls
#Depends on:
#Arguments: Ycase: number of 1's in cases
#            Ycontrol: number of 1's in controls.
#            caseN: Number of cases;
#            controlN: Number of controls.
#Details:
#Value: LRT statistic
#Note:
#Author(s): ZWu, DZhang
#References: Chen, ..., Karchin 2013 PloS Paper
#See Also:
#Example:
#####
LRTStat.condi <- function(Ycase, Ycontrol, caseN, controlN)
{
    pA = (Ycase + 1)/(caseN + 2);
    pU = (Ycontrol + 1)/(controlN + 2);
    p = (Ycase+ Ycontrol + 2)/(caseN+ controlN+ 4);

    if (pA >= pU)
    {

```

```

        statistic = Ycase*log10(pA)+(caseN-Ycase)*log10(1-pA)+Ycontrol*log10(pU)+(controlN-
Ycontrol)*log10(1-pU)-(Ycase+Ycontrol)*log10(p)-(caseN+controlN-Ycase-Ycontrol)*log10(1-p);
    }
    else
    {
        statistic = Ycase*log10(pU)+(caseN-Ycase)*log10(1-pU)+Ycontrol*log10(pA)+(controlN-
Ycontrol)*log10(1-pA)-(Ycase+Ycontrol)*log10(p)-(caseN+controlN-Ycase-Ycontrol)*log10(1-p);
    }
    return(statistic);
}

```

weightBurdenTest_invDir_ToGetTypeIErrorCurve.R

```

# BEGINCONF
# [geneSize]
# [permuN]
# [pvalue]
# ENDCONF

```

#Note this function give the inversed direction weight, contrasting to weightBurdenTest.R

#####

#Description: Interface R function to be called by Variant Associate Tools. It has the choice to calculate an association test statistic based on direction weighted burden score. That is, if MAF in cases < MAF in controls, direction weight = 1, o.w., =-1. Further explicit weights can be multiplied by these direction weights to get the final weights. Permutation test is used to get p-values. Direction weights are re-calculated after each permutation.

#Depends on:

#Arguments: dat: data from Variant Tools. Require: (a) For phenotype, 1s indicate cases; 0s indicate controls. (b) genotype scores are counts of the minor alleles (mutations).

Yphenotype.name: Name of the phenotype

testFunc: The name of the association test statistic calculating function

weightScheme: "noWei": no weighting; "expWei": explicit weight only; "dirWei": direction weight only; "expDirWei": explicit weights are multiplied b direction weights to get the final weights. Default is no weighting. Variant Tools command must specify the expliciate weight for "expWei" and "expDirWei";

permuN: Number of the permutations

#Details:

#Value: Size of snp group; # of permutations; pvalue

#Note:

#Author(s): ZWu, DZhang

#References:

#See Also:

#Example:

#####

```

weightBurdenTest_invDir_ToGetTypeIErrorCurve = function (dat, phenotype.name = "HTN",
testFunc="LRTStat", weightScheme="noWei", permuN = 5000) {

  phenotypes = dat@Y[,phenotype.name]; #phenotype
  phenotypes = sample(phenotypes); ##To get type I error even for functional genes

  caseN = sum(phenotypes==1);
  controlN = sum(phenotypes==0);

  genotypes = dat@X; #Genotype
  genotypes[is.na(genotypes)] = 0; #Missing genotype are treated as the major allele.
  #genotypes= round(genotypes);

  #calculate direction weights based on genotypes and phenotypes
  if (weightScheme %in% c("noWei", "dirWei")) weights = data.frame(rep(1, ncol(genotypes)));
#no weighting
  if (weightScheme %in% c("expWei", "expDirWei")) weights = dat@V;
  #explicit weights
  if (weightScheme %in% c("dirWei", "expDirWei")) {
    DirectionWeight = unlist(apply(genotypes, 2, function(g) {
      maf.case = sum(g[phenotypes==1])/(2*caseN);
      maf.contr = sum(g[phenotypes==0])/(2*controlN);
      #ifelse(maf.case >= maf.contr, 1, -1); #set up in weightBurdenTest.R
      ifelse(maf.case >= maf.contr, -1, 1);
    }));
    weights = weights*DirectionWeight;
  }

  #For real data

  stat = do.call(testFunc, list(phenotypes, genotypes, weights, caseN, controlN));

  #For permutation test

  stat.perms = 0;

  for (i in 1:(permuN/10)) {
    pheno.perm = sample(phenotypes);
    #Permutation stage 1
  }
}

```

```

        if (weightScheme %in% c("noWei", "dirWei")) weights = data.frame(rep(1,
ncol(genotypes)));
        if (weightScheme %in% c("expWei", "expDirWei")) weights = dat@V;

        if (weightScheme %in% c("dirWei", "expDirWei")) {
            DirectionWeight = unlist(apply(genotypes, 2, function(g) {
                maf.case = sum(g[pheno.perm==1])/(2*caseN);
                maf.contr = sum(g[pheno.perm==0])/(2*controlN);
                #ifelse(maf.case >= maf.contr, 1, -1); #set up in weightBurdenTest.R
                ifelse(maf.case >= maf.contr, -1, 1);
            }));
            weights = weights*DirectionWeight;
        }
        stat.perms[i] = do.call(testFunc, list(pheno.perm, genotypes, weights, caseN, controlN));
    }

    pValue = (sum(stat.perms >= stat)+0.5)/(length(stat.perms)+1);

    if (pValue > 0.1) {
        return (list(geneSize=ncol(genotypes), permuN=i, pvalue=pValue));
    }else {
        for (i in (permuN/10+1):permuN) {
            #Permutation stage 2

            pheno.perm = sample(phenotypes);
            if (weightScheme %in% c("noWei", "dirWei")) weights = data.frame(rep(1,
ncol(genotypes)));
            if (weightScheme %in% c("expWei", "expDirWei")) weights = dat@V;

            if (weightScheme %in% c("dirWei", "expDirWei")) {
                DirectionWeight = unlist(apply(genotypes, 2, function(g) {
                    maf.case = sum(g[pheno.perm==1])/(2*caseN);
                    maf.contr = sum(g[pheno.perm==0])/(2*controlN);
                    #ifelse(maf.case >= maf.contr, 1, -1); #set up in weightBurdenTest.R
                    ifelse(maf.case >= maf.contr, -1, 1);
                }));
                weights = weights*DirectionWeight;
            }
            stat.perms[i] = do.call(testFunc, list(pheno.perm, genotypes, weights, caseN,
controlN));
        }

        pValue = (sum(stat.perms >= stat)+0.5)/(length(stat.perms)+1);
    }

```

```

        return (list(geneSize=ncol(genotypes), permuN=i, pvalue=pValue));
    }
}

```

```

#####
#Description: Calculate the LRT statistic according to Bernoulli distribution. The burden score can be
weighted. The LRT statistic is gotten at the best burden score threshold that maximizes the statistic.
#Depends on:
#Arguments: phenotypes: A vector of binary phenotype values. Require: 1s indicate cases; 0s indicate
controls.
#           genotypes: Genotype data frame (cols: variants; rows: sample subjects)
#           weights: A one-col data frame vector of variant weights wrt to genotypes. Default: all
1s (no weighting).
#           caseN: Number of cases;
#           controlN: Number of controls.
#Details:
#Value: LRT statistic
#Note:
#Author(s): ZWu, DZhang
#References: Chen, ..., Karchin 2013 PloS Paper
#See Also:
#Example:
#####
LRTStat <- function(phenotypes, genotypes, weights=data.frame(rep(1, ncol(genotypes))), caseN,
controlN)
{
    burdenScores = data.matrix(genotypes) %*% data.matrix(weights);

    thresholds = c(unique(burdenScores), max(burdenScores)+1);

    stats = array(0, dim=c(length(thresholds)));

    for (i in 1:length(thresholds ))
    {

```



```

        Ys = (burdenScores >= thresholds[i]);

        Ycase = sum(Ys[phenotypes==1]);

        Ycontrol = sum(Ys[phenotypes==0]);

        stats[i] = LRTStat.condi(Ycase, Ycontrol, caseN, controlN);

    }

    finalStat = max(stats[!is.na(stats)]);

    return(finalStat);
}

```

```

#####
#Description: Calculate the LRT statistic, conditional on given numbers of 1's in cases and controls
#Depends on:
#Arguments: Ycase: number of 1's in cases
#           Ycontrol: number of 1's in controls.
#           caseN: Number of cases;
#           controlN: Number of controls.
#Details:
#Value: LRT statistic
#Note:
#Author(s): ZWu, DZhang
#References: Chen, ..., Karchin 2013 PloS Paper
#See Also:
#Example:
#####

```

```

LRTStat.condi <- function(Ycase, Ycontrol, caseN, controlN)
{
    pA = (Ycase + 1)/(caseN + 2);

    pU = (Ycontrol + 1)/(controlN + 2);

    p = (Ycase+ Ycontrol + 2)/(caseN+ controlN+ 4);

    if (pA >= pU)

```

```

    {
        statistic = Ycase*log10(pA)+(caseN-Ycase)*log10(1-pA)+Ycontrol*log10(pU)+(controlN-
Ycontrol)*log10(1-pU)-(Ycase+Ycontrol)*log10(p)-(caseN+controlN-Ycase-Ycontrol)*log10(1-p);
    }
else
{
    statistic = Ycase*log10(pU)+(caseN-Ycase)*log10(1-pU)+Ycontrol*log10(pA)+(controlN-
Ycontrol)*log10(1-pA)-(Ycase+Ycontrol)*log10(p)-(caseN+controlN-Ycase-Ycontrol)*log10(1-p);
}
return(statistic);
}

```

direct structure based prediction results of the effect of snv on PPI

P00533	P00533	EGFR	EGFR	A	I	3b2u	line8504	A
P00533	P00533	EGFR	EGFR	A	B	3njp	line8504	A
P00749	Q03405	PLAU	PLAUR	A	U	3bt2	line20283	U
P01040	P01040	CSTA	CSTA	A	B	1n9j	line5181	A
P01040	P07711	CSTA	CTSL	D	A	3kse	line5181	D
P01040	P07858	CSTA	CTSB	C	A	3k9m	line5181	C
P01133	P00533	EGF	EGFR	B	A	1nql	line8504	A
P01343	P08833	IGF1	IGFBP1	C	H	2dsq	line8410	H
P02730	P02730	SLC4A1	SLC4A1	P	Q	1hyn	line17387	P
P06865	P07686	HEXA	HEXB	A	B	2gk1	line15611	A
P09467	P09467	FBP1	FBP1	A	B	3kbz	line10359	A
P20591	P20591	MX1	MX1	A	B	3ljb	line21431	A

P20827	P29317	EFNA1	EPHA2	B	A	3mbw	line2347	B
P29474	P29474	NOS3	NOS3	A	B	1m9m	line9663	A
P35228	P35228	NOS2	NOS2	A	B	1nsi	line16938	A
P36404	Q9Y2Y0	ARL2	ARL2BP	A	B	3doe	line12905	A
Q9H2X3	Q9H2X3	CLEC4M	CLEC4M	A	B	1k9j	line19024	A

Hypertension related gene list (based on literature)

ABCA3	ACSM3	ACVRL1	ADC	ADD1	ADRA1A
AGT	AGTR1	ATP1B1	ATP2B1	BMPR2	CACNB2
CAV1	CLCNKB	CPS1	CSK	CUL3	CYP11B1
CYP11B2	CYP17A1	CYP3A5	ECE1	EDN3	FGF5
GNAS	GNB3	HFE	HSD11B2	KCNJ1	KCNK3
KCNMB1	KLHL3	LSP1	MADH9	MECOM	MTHFR
NOS2	NOS2A	NOS3	NPR3	NR3C2	PLEKHA7
PNMT	PTGIS	RET	RETN	RGS5	SARS2
SCNN1B	SCNN1G	SDHA	SDHAF2	SDHB	SDHC
SDHD	SELE	SH2B3	SLC12A1	SLC12A3	SMAD9
SOX6	TBX3	TBX5	TNNT3	ULK4	UMOD
VHL	WNK1	WNK4	ZNF652		

Hypertension centered network- interactions from HINT database

A2M	NOS3	NOS3	EFEMP2
ACTB	SMAD9	NOS3	EXOC6
ACTN2	NOS3	NOS3	FIS1
ACTN4	NOS3	NOS3	HTRA1
ACVR1	BMPR2	NOS3	IMMT
ACVR2A	ACVRL1	NOS3	KANK2
ACVRL1	ACVRL1	NOS3	MAST1
ACVRL1	ENG	NOS3	MPRIP
ACVRL1	TGFB1	NOS3	NOS3
ACVRL1	TGFBR2	NOS3	PI4K2A
ADD1	C1orf109	NOS3	PPP2R5A
ADD1	CSNK2B	NOS3	RNF31
ADD1	GEMIN7	NOS3	RNF32
ADD1	HMG20A	NOS3	ST13
ADD1	MAP1LC3B	NOS3	TXNDC11
ADD1	PRKCD	NOS3	UMPS
ADD1	SPTA1	NPPB	NPR3
ADD1	TK1	NPPC	NPR3
AES	TBX3	NR3C2	EIF3I
AGT	ACE2	NR3C2	NCOA1

AGT	AGT	NR3C2	PROX1
AGT	CTSG	NR3C2	PSMC5
AGT	EWSR1	NR3C2	SRC
AGT	GRB2	NR3C2	TRIM24
AGT	PRCP	OAZ3	ADC
AGTR1	PDE6H	PICK1	PLEKHA7
AGT	REN	PLCG1	RET
AKT1	NOS3	PNMT	KLHL8
AP2A1	SMAD9	PPL	PLEKHA7
ARHGDI1A	SDHC	PRKAR1A	WNK1
ARNT	SMAD9	PRKAR1B	WNK1
ATP1B1	ATP1B1	RANBP9	WNK1
ATP1B1	CRIP2	RAP1GAP2	WNK1
ATP1B1	DDAH2	RASAL3	WNK1
ATP1B1	EZH2	RET	FRS2
ATP1B1	LRIF1	RET	NOTCH2NL
ATP1B1	MLL4	RET	RET
ATP1B1	PSME1	RET	SHC1
ATP1B1	RIF1	RET	SHC3
ATP1B1	TRMT2A	RET	STAT3
ATP2B1	ATP2B1	SCNN1B	NEDD4L
ATP8	WNK1	SCNN1B	STX1A
ATXN1	WNK1	SCNN1B	WWP2
B2M	HFE	SCNN1G	NEDD4L
BAD	WNK1	SCNN1G	STX1A
BAG3	WNK1	SDHAF2	CCDC90B
BDKRB2	NOS3	SDHAF2	SDHAF2
BMP2	BMPR2	SDHB	PIK3R5
BMP4	BMPR2	SDHB	SPRY2
BMP7	BMPR2	SDHC	SDHC
BMPR2	BMPR2	SELE	SELPLG
BMPR2	GDF5	SH2B3	SH2B3
BMPR2	MAPK8	SH2B3	USP32P1
BMPR2	TGFBR1	SLC9A1	WNK1
CA8	TBX3	SMAD2	SMAD9
CACNB2	CTBP2	SMAD3	SMAD9
CACNB2	HEXIM2	SMAD4	SMAD9
CACNB2	PRKACA	SMAD9	ABTB1
CAV1	CAV1	SMAD9	AFF1
CAV1	CAV2	SMAD9	ARHGAP9
CAV1	CSK	SMAD9	ARID1B
CAV1	EGFR	SMAD9	ASB2
CAV1	PLD1	SMAD9	ASH2L
CAV1	PTPN1	SMAD9	BAZ1A
CAV1	SRC	SMAD9	C10ORF2
CAV1	TRAF2	SMAD9	CAMSAP1
CBL	RET	SMAD9	CEP135
CCNDBP1	PLEKHA7	SMAD9	CHPF
CENPJ	WNK1	SMAD9	CLPB
CHAF1A	WNK1	SMAD9	CPXM2
CLK2	ECE1	SMAD9	CSRP3
CLTC	WNK1	SMAD9	CTR9
CNTN1	SCNN1B	SMAD9	CXXC5

COPS5	WNK1	SMAD9	DIAPH3
CSDE1	WNK1	SMAD9	DKK1
CSH1	SMAD9	SMAD9	DNAJA3
CSK	DOK3	SMAD9	DNAJC7
CSK	FGR	SMAD9	DSTN
CSK	FYN	SMAD9	EIF3C
CSK	KANK4	SMAD9	EIF3F
CSK	LCK	SMAD9	ERVV-1
CSK	PTPRB	SMAD9	EVC2
CSK	PTPRC	SMAD9	EXPH5
CSK	PTPRG	SMAD9	HEY1
CSK	PTPRJ	SMAD9	HEYL
CSK	PTPRK	SMAD9	HUWE1
CSK	PTPRO	SMAD9	KDM6A
CSK	PTPRZ1	SMAD9	KIAA0226
CSK	PXN	SMAD9	LEMD3
CSK	RGS16	SMAD9	LMO4
CSK	SHC1	SMAD9	MAN1A2
CSK	SRC	SMAD9	MAN1C1
CSNK2B	SDHA	SMAD9	MAN2B1
CTBP1	MECOM	SMAD9	MBD1
CUL3	ABTB1	SMAD9	MCM3AP
CUL3	ABTB2	SMAD9	METAP1
CUL3	BTBD6	SMAD9	MGAT1
CUL3	GMCL1	SMAD9	MIA3
CUL3	KCTD13	SMAD9	MLL2
CUL3	KCTD6	SMAD9	MLL4
CUL3	KCTD7	SMAD9	MOGAT1
CUL3	KCTD9	SMAD9	MTCH1
CUL3	KEAP1	SMAD9	MTMR10
CUL3	KLHL12	SMAD9	MTMR11
CUL3	KLHL2	SMAD9	NAGK
CUL3	KLHL3	SMAD9	OTUB1
CUL3	ZMAT4	SMAD9	PABPC4
CYP11A1	SMAD9	SMAD9	PAPPA
DOK1	RET	SMAD9	PELP1
DST	SMAD9	SMAD9	PHKA2
E4F1	SMAD9	SMAD9	PIR
ECE1	EDN1	SMAD9	PKP2
ECE1	KRTAP10-1	SMAD9	PLEC
ECE1	KRTAP10-3	SMAD9	PNPLA2
ECE1	KRTAP10-5	SMAD9	PPARD
ECE1	KRTAP10-8	SMAD9	PPP2R5E
ECE1	KRTAP1-3	SMAD9	PSAP
EDN3	EDNRB	SMAD9	PSMD8
EEF1A1	SDHAF2	SMAD9	QARS
EFEMP1	NOS3	SMAD9	RANBP9
EGFR	SH2B3	SMAD9	RFX1
EIF3A	WNK1	SMAD9	RMND5A
EIF3E	SMAD9	SMAD9	RNF123

ELAVL1	NOS3	SMAD9	RRBP1
ELAVL3	NOS3	SMAD9	SECISBP2
EPB41L2	WNK1	SMAD9	SF3B1
EPRS	SMAD9	SMAD9	SIL1
ERBB2	SH2B3	SMAD9	SMAD9
ERBB3	SH2B3	SMAD9	SMAP1
ESR2	WNK4	SMAD9	SMG1
EWSR1	KCNMB1	SMAD9	SNRNP70
FAM13B	WNK1	SMAD9	SPTBN1
FAM53C	WNK1	SMAD9	STAG1
FAM65B	WNK1	SMAD9	SVEP1
FAM82A2	WNK1	SMAD9	TBCD
FGF5	FGFR3	SMAD9	TERF1
FHL3	WNK1	SMAD9	TINAGL1
FHOD1	WNK1	SMAD9	TMEM57
FLI1	SMAD9	SMAD9	TRIM29
FLII	SMAD9	SMAD9	TRIP12
FLNA	VHL	SMAD9	TTC37
FLNC	WNK1	SMAD9	UBA6
FN1	SMAD9	SMAD9	UBE3A
FN1	VHL	SMAD9	UBQLN1
FTL	SMAD9	SMAD9	UBQLN4
GCDH	NOS3	SMAD9	UNC45A
GEM	SDHB	SMAD9	VCPIP1
GNB3	GNG3	SMAD9	VPS8
GNB3	GNG4	SMAD9	XAB2
GNB3	TGFBR1	SMAD9	YWHAQ
GOLGA2	NOS3	SMAD9	ZEB2
GPR25	RETN	SMAD9	ZNF484
GRB2	SH2B3	SMAD9	ZNF557
GRN	SMAD9	SMAD9	ZNF587
HDAC9	WNK1	SMAD9	ZNF592
HFE	HFE	SMAD9	ZNF8
HFE	TFR2	SMAD9	ZNF83
HFE	TFRC	SMAD9	ZSCAN4
HIF1A	VHL	SSFA2	WNK1
HIVEP2	WNK1	STK39	WNK4
HSP90AA1	NOS3	SYNPO	WNK1
IKBKAP	SDHAF2	TBX3	PLEKHF2
IMMT	SDHAF2	TBX3	PRR20A
KCNK3	YWHAB	TBX3	TOLLIP
KCNK3	YWHAZ	TBX5	BAIAP2
KEAP1	KLHL3	TBX5	ZMYND10
KIAA0232	WNK1	TCEB1	VHL
KLHL3	C6orf165	TCEB2	VHL
KLHL3	KLHL12	TNFAIP1	CUL3
KLHL3	KLHL3	TNFAIP3	WNK1
LCK	SH2B3	TNNI2	TNNT3
LCP2	WNK1	TNNT3	HAP1
LNPEP	SMAD9	TNNT3	NUDT3
LRMP	WNK1	TNNT3	SNUPN
LRP5	SMAD9	TNNT3	TNNT3
MADD	WNK1	TNNT3	TSG101

MAGEA11	NOS3	TNRC6A	WNK1
MAP2K1	WNK1	TNS1	WNK1
MDF1	VHL	TSC22D1	WNK1
MECOM	MECOM	ULK4	SMURF1
MECOM	SMAD3	VBP1	VHL
MEOX2	RGS5	VHL	CUL2
MORF4L2	KLHL3	VHL	ZNF512B
MPHOSPH9	WNK1	WNK1	C5ORF25
MSL2	WNK1	WNK1	CGNL1
MTHFR	NAA38	WNK1	SYNPO2
MYH9	WNK1	WNK1	WNK1
NEDD4L	WNK1	WNK1	WNK2
NEDD4	SCNN1B	WWTR1	WNK1
NEDD4	SCNN1G	XRCC6	SDHC
NEDD8	CUL3	YWHAE	WNK1
NOS2	NOS2	YWHAG	WNK1
NOS2	RAC1	YWHAZ	WNK1
NOS2	SLC9A3R1	ZFP106	WNK1
NOS3	CDC37	ZNF839	WNK1
		ZYX	WNK1

Structurally Resolved Hypertension Centered Network through INstruct database

BMP7	BMPR2	SMAD9	SMAD9
NOS3	NOS3	FYN	CSK
SMAD3	SMAD9	ACVRL1	ACVRL1
SH2B3	ERBB3	RET	SHC1
ACVRL1	ACVR2A	SMAD9	SMAD9
SMAD9	SMAD2	SRC	CSK
ACVRL1	ACVR2A	SRC	CSK
SMAD9	SMAD4	SH2B3	SH2B3
SMAD9	SMAD2	GNB3	GNG3
PTPRB	CSK	ACVRL1	ACVRL1
NOS2	NOS2	STAT3	RET
TFR2	HFE	SHC1	CSK
SRC	CSK	PLCG1	RET
PTPRC	CSK	RET	RET
FGR	CSK	PTPRJ	CSK
NOS3	NOS3	PTPRB	CSK
CSK	PTPRO	TGFBR1	BMPR2
ACVR1	BMPR2	MECOM	MECOM
CSK	PXN	SRC	CSK
HFE	HFE	FYN	CSK
ATP2B1	ATP2B1	LCK	CSK
TCEB1	VHL	WNK1	WNK1
SH2B3	GRB2	PTPRB	CSK
NPPC	NPR3	ATP2B1	ATP2B1
PTPRC	CSK	TGFBR1	BMPR2
FGF5	FGFR3	SRC	CSK
SDHAF2	SDHAF2	NOS2	NOS2
BMP4	BMPR2	HFE	HFE

ACVR1	BMPR2	NEDD8	CUL3
NOS3	NOS3	NOS2	NOS2
AGT	CTSG	PTPRJ	CSK
FGR	CSK	PTPRJ	CSK
GNB3	GNG4	PTPRG	CSK
RET	RET	PTPRG	CSK
SRC	CSK	TGFBR1	BMPR2
HFE	HFE	FGR	CSK
NOS2	NOS2	NOS2	NOS2
FGR	CSK	SRC	CSK
AGT	AGT	PTPRZ1	CSK
WNK1	PRKAR1B	FGR	CSK
PTPRZ1	CSK	LCK	CSK
FYN	CSK	SH2B3	GRB2
ATP1B1	ATP1B1	LCK	CSK
FYN	CSK	ATP2B1	ATP2B1
EGFR	SH2B3	SH2B3	SH2B3
BMP2	BMPR2	NOS3	NOS3
LCK	CSK	PTPRZ1	CSK
FGR	CSK	SHC1	CSK
HFE	TFRC	SRC	CSK
FYN	CSK	FGR	CSK
FYN	CSK	PTPRC	CSK
SRC	CSK	FYN	CSK
NOS3	NOS3	WNK1	MAP2K1
LCK	CSK	BMPR2	BMPR2
NOS3	NOS3	FYN	CSK
B2M	HFE	SDHC	SDHC
SH2B3	LCK	ERBB2	SH2B3
SH2B3	SH2B3	CSK	PTPRO
ACVRL1	ACVRL1	ATP2B1	ATP2B1
PTPRG	CSK	FGF5	FGFR3
ATP2B1	ATP2B1	ATP2B1	ATP2B1
TNNI2	TNNT3	SHC1	CSK
ATP2B1	ATP2B1	ATP2B1	ATP2B1
ACVRL1	ACVR2A	NOS2	NOS2
NOS2	NOS2	LCK	CSK
B2M	HFE	VHL	HIF1A
ACVR1	BMPR2	AGT	REN
SH2B3	LCK	NOS3	NOS3
WNK1	PRKAR1A	WNK1	WNK2
FGR	CSK	WNK4	STK39
LCK	CSK	FGR	CSK
LCK	CSK	SMAD9	SMAD4
CSK	PTPRO	MAPK8	BMPR2
SH2B3	LCK	SMAD3	SMAD9
PLCG1	RET	BMPR2	BMPR2
LCK	CSK	SH2B3	LCK
TNNT3	TNNT3	SH2B3	GRB2
SHC3	RET	FYN	CSK
		NPPB	NPR3

