1. Study overview

A screenshot of a computer

Description automatically generated

1. Data Normalization and Integration
2. DE analysis (chair)
3. 1-cosine similarity
4. Results
   1. Drug vs disease
   2. Drug vs drug
5. Data Summary
   1. Data availability
6. References
   1. Deshpande, Raamesh et al. “Comparison of profile similarity measures for genetic interaction networks.” PloS one vol. 8,7 e68664. 10 Jul. 2013, doi:10.1371/journal.pone.0068664
7. Study Overview
   * What is the main objective of your study?
     1. The main objective was to implement the characteristic direction statistical method on drug and disease data, finding the connections of gene expression movement, regardless of the direction between a given drug and kidney disease state.
   * What are the key methods and techniques you employed?
     1. I’ve used various methods from other papers.
     2. After that, I processed the drug data as such:
        1. Loaded the data from either GSE local files or from queries.
        2. Filtered, cleaned, and normalized data in fpkm.
        3. Standardized genes for multi-study integration and preparation for chdir function
   * What is the significance of your study in the field?
     1. Identifying differentially expressed genes (DEG) is a fundamental step in studies that perform genome wide expression profiling. Typically, DEG are identified by univariate approaches such as Significance Analysis of Microarrays (SAM) or Linear Models for Microarray Data (LIMMA) for processing cDNA microarrays, and differential gene expression analysis based on the negative binomial distribution (DESeq) or Empirical analysis of Digital Gene Expression data in R (edgeR) for RNA-seq profiling.. The application of the Characteristic Direction method may shed new light on relevant biological mechanisms that would have remained undiscovered by the current state-of-the-art DEG methods. Significance is establishing the use of a relatively novel method for drug and disease gene expression comparisons. We also find a few drug-disease interactions that might be worth investigating. The information analyzed can compare gene movement of drugs in different fields of medicine and assesses the clash or harmony of gene movement between the two drugs or drug and disease pair.
8. Data Normalization and Integration
   * What datasets did you use for your study?
     1. For the disease data (Tajti, Ferenc et al. “A Functional Landscape of CKD Entities From Public Transcriptomic Data.” Kidney international reports vol. 5,2 211-224. 13 Nov. 2019, doi:10.1016/j.ekir.2019.11.005):
        1. Data Collection
        2. The raw data CEL files of each microarray dataset, GSE20602, 10 GSE32591, 11 GSE37460, 11 GSE47183, 12, 13, and GSE50469, 14, were downloaded and imported to R (R version 3.3.2). For more information, see the Supplementary Methods.
        3. Normalization
        4. Cyclic loess normalization was applied using the limma package.15, 16, 17 YuGene transformation was carried out using the YuGene R package.18
        5. Batch Effect Mitigation: Method
        6. First, we structured the data in a platform-specific manner. Then, we conducted differential gene expression analysis between the identical biological conditions from distinct study sources after cyclic loess normalization. We subsequently removed the genes that are significantly differentially expressed between them, as it indicated differences mainly due to the data source, rather than the biological difference. We applied this procedure for the data fragments coming from Affymetrix (Santa Clara, CA) Human Genome U133 Plus 2.0 Array and Affymetrix Human Genome U133A Array. Next, we merged the data sets between the 2 platforms using the overlapping genes, followed by a process to mitigate the platform-induced batch effect. This latter procedure is similar to the one used for the data source–specific batch effect mitigation.
     2. For drug data (multiple datasets, so I just put the list of processed file names here. Write down the unique list of GSE geodatasets from this list here and within those datasets, write down the drugs taken from them. The format is chdir\_ drug name \_ GEO Dataset \_ results.txt):
        1. chdir\_Atorvastatin\_GSE196701\_results.txt
        2. chdir\_Bicarbonate\_GSE200638\_results.txt
        3. chdir\_Canagliflozin\_GSE106156\_results.txt
        4. chdir\_Candesartan\_GSE59913\_results.txt
        5. chdir\_Cortisone\_GSE59913\_results.txt
        6. chdir\_Fineronone\_GSE183841\_results.txt
        7. chdir\_Furosemide\_GSE59913\_results.txt
        8. chdir\_Hydrocortisone\_GSE59913\_results.txt
        9. chdir\_Ibuprofen\_GSE59913\_results.txt
        10. chdir\_JNJ39933673\_GSE199437\_results.txt
        11. chdir\_Lisinopril\_GSE199437\_results.txt
        12. chdir\_losartan\_GSE159059\_results.txt
        13. chdir\_Methotrexate\_GSE59913\_results.txt
        14. chdir\_MMF\_GSE153021\_results.txt
        15. chdir\_Naproxen\_GSE59913\_results.txt
        16. chdir\_Prednisolone\_GSE59913\_results.txt
        17. chdir\_Prednisone\_GSE153021\_results.txt
        18. chdir\_ramipril\_GSE226353\_results.txt
        19. chdir\_Sparsentan\_GSE225447\_results.txt
        20. chdir\_Spironolactone\_GSE59913\_results.txt
        21. chdir\_Valsartan\_GSE59913\_results.txt
   * How did you normalize the data to ensure consistency and comparability?
     1. I normalized the data by focusing on choosing all kidney data first, then normalized most data through DESeq2. Some datasets had DESeq2 Processed data, and as such, I did not do this step again for them. After this, I converted the gene IDs to gene symbols that are readable to the chdir program, as well as letting us standardize the genes for data combination.
   * What techniques did you use to integrate data from different sources?
     1. When we come to this part, ask for the code and we can specifically write the portion for this.
   * Did you encounter any challenges during data normalization and integration?
     1. The data was quite messy for our purpose, as it was originally intended for other uses. The disease state data was rather clean and easy to manage, since they came from the same tissue sources.
     2. The drug data was all over the place, as I used many sources to get the expression data. These multiple sources all had unique formats for storing expression data, meta data, and phenotype data.
        1. Sifting through this to get what I wanted was difficult and required unique methods per data set.
9. DE Analysis
   * What is chdir and how did you apply it in your study?
     1. Chdir is the scripts nickname for characteristic direction. Let’s avoid calling it chdir, and let’s stick to characteristic direction as it sound better. Here is the math and explanation behind characteristic direction:

### Computing the characteristic direction and identifying differentially expressed genes

Classification approaches, for example those that predict clinical outcome from gene expression data, are inherently multivariate as they use the structure of the gene expression profiles as a whole in order to distinguish between biological conditions or classes. Our approach is to repurpose linear classification methods in order to characterize differential expression and identify DEG. We use a linear classification scheme, which defines a separating hyper-plane; the orientation of which we show can be interpreted to identify DEG. We also find that the direction normal to the separating hyper-plane provides a simple geometrical conceptualization of the differential expression, which naturally leads to extensions of the approach, such as a new formulation of gene set enrichment analysis.

Suppose we have gene expression data from a number of samples N, in which the expression of p genes is measured, and then let each expression profile sample form a row of the matrix **X** (a N × p matrix). For generality at this point we shall consider the case where each of the expression samples comes from one of K classes belonging to the set G. In linear discriminant analysis (LDA) the log-ratio of class posteriors P (G|X), is written as follows (see Additional file [1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4000056/#S1) for a derivation),

*𝑙𝑜𝑔*Pr(*𝐺*=*𝑘*|*𝑋*=*𝑥*)Pr(*𝐺*=*𝑙*|*𝑋*=*𝑥*)=*𝑙𝑜𝑔𝜋𝑘𝜋𝑙*−12(*𝜇𝑘*−*𝜇𝑙*)*𝑇*Σ−1(*𝜇𝑘*−*𝜇𝑙*)+*𝑥𝑇*Σ−1(*𝜇𝑘*−*𝜇𝑙*)

(1)

where, πk, is the class mean, and it is assumed that both classes have the same covariance matrix, Σ. Then the orientation of the separating hyper-plane (between classes k and l) is defined by the normal p-vector, in the third term on the right hand side, that we label b,

b = Σ-1(μk - μl).

(2)

The estimation, from the data, of the terms in this equation is explained in the Additional file [1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4000056/#S1). Below we will interpret the direction of the p-vector, b, as the direction in expression space that best characterizes the differential expression, and show how the components of this vector can be used to identify differentially expressed genes. However, first we note a few potential issues: the calculation involves the inverse of a very large p × p matrix which is not only expensive to compute but also the elements must be estimated from a relatively small sample-size (p >>N), which means that the matrix is singular and this leads to large variance in the results even when using the generalized inverse.

The issue of singularity and large variance can be tackled with a regularization procedure, for example, the covariance matrix can be shrunk to the scalar variance as follows,

Σˆ(*𝛾*)=*𝛾*Σˆ+(1−*𝛾*)*𝜎*2*𝐼𝑝*,withγϵ[0,1]

(3)

where *𝛴*ˆ is the estimated covariance matrix, and σ2 is the scalar covariance (see Additional file [1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4000056/#S1) for elaboration). The inclusion of a constant on the diagonal resolves the singularity problem, and the modulation of the off-diagonal terms helps to reduce noise arising from the estimation of covariance from few samples.

The problem of computational expense is efficiently overcome with the singular-value decomposition trick [[26](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4000056/#B26)-[28](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4000056/#B28)] which also admits a solution in the limit of zero shrinkage by working in the subspace spanned by the data, rather than the full expression space (see the Additional file [1](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4000056/#S1)). The normalized vector *𝑏*ˆ contains only information about the direction of the normal to the separating hyper-plane. The components of *𝑏*ˆ are the direction cosines, and their magnitude quantifies the degree of alignment of the direction to axes corresponding to each gene. The sign of each component can be interpreted as the sign of the contribution of each gene to the differential expression. Another way to picture this interpretation of gene significance is to consider the identity,

∑*𝑖*=1*𝑝𝑏*ˆ2*𝑖*≡1

(4)

Then the contribution of each *𝑏*ˆ21 to this sum can be interpreted as quantifying the relative contribution of each component to the total differential expression giving the significance of the corresponding gene. The above interpretation provides a quantitative measure of the relative, but not absolute, significance of each gene to the differential expression, and as such can be used to rank the genes in order of significance. However, we also want to identify a shortlist of significant DEGs. This could be done completely within the framework we have outlined by using a L1 regularization scheme in place of that used in the shrinkage equation above; such a penalty results in automatic feature selection because many components fall to zero; the genes corresponding to the features retained would then comprise the DEGs. An alternative method to deriving a significance threshold is described below.

Here is the code version that I implemented:

"chdir" <-

function(ctrl,expm,genes,r=1)

# This function caclulates the characteristic direction for a gene expression dataset.

# ctrl: control gene expressoion data, a matrix object

# expm: experiment gene expression data, a matrix object

# b: return value, a vector of n-components, representing the characteristic

# direction of the gene expression dataset. n equals to the number of genes in the

# expression dataset. b is also a matrix object. b is sorted by its components'

# absolute values in descending order.

# r: regularized term. A parameter that smooths the covariance matrix and reduces

# potential noise in the dataset. The default value for r is 1, no regularization.

#

# For the input matrix rows are genes and columns are gene expression profiles.

# r is the regulization term ranging [0,1]. b is the characteristic direction.

# ctrl(control) and expm(experiment) matrices should have the same number

# of genes(rows).

#

# Author: Qiaonan Duan

# Ma'ayan Lab, Icahn School of Medicine at Mount Sinai

# Jan.13, 2014

#

# Add gene symbols to results. Apr. 4, 2014

{

if(dim(ctrl)[1]!=dim(expm)[1]){

stop('Control expression data must have equal number of genes as experiment expression data!')

}

if(any(is.na(ctrl))||any(is.na(expm))){

stop('Control expression data and experiment expression data have to be real numbers. NA was found!')

}

# There should be variance in expression values of each gene. If

# gene expression values of a gene are constant, it would dramatically

# affect the LDA caculation and results in a wrong answer.

constantThreshold <- 1e-5;

ctrlConstantGenes <- diag(var(t(ctrl))) < constantThreshold

expmConstantGenes <- diag(var(t(expm))) < constantThreshold

if (any(ctrlConstantGenes)){

errMes <- sprintf('%s row(s) in control expression data are constant. Consider Removing the row(s).',paste(as.character(which(ctrlConstantGenes)),collapse=','))

stop(errMes)

}else if(any(expmConstantGenes)){

errMes <- sprintf('%s row(s) in experiment expression data are constant. Consider Removing the row(s).',paste(as.character(which(expmConstantGenes)),collapse=','))

stop(errMes)

}

# place control gene expression data and experiment gene expression data into

# one matrix

combinedData <- cbind(ctrl,expm)

# get the number of samples, namely, the total number of replicates in control

# and experiment.

dims <- dim(combinedData)

samplesCount <- dims[2]

# the number of output components desired from PCA. We only want to calculate

# the chdir in a subspace that capture most variance in order to save computation

# workload. The number is set 20 because considering the number of genes usually

# present in an expression matrix 20 components would capture most of the variance.

componentsCount <- min(c(samplesCount-1,20))

# use the nipals PCA algorithm to calculate R, V, and pcvars. nipals algorithm

# has better performance than the algorithm used by R's builtin PCA function.

# R are scores and V are coefficients or loadings. pcvars are the variances

# captured by each component

pcaRes <- nipals(t(combinedData),componentsCount,1e5,1e-4)

R <- pcaRes$T

V <- pcaRes$P

pcvars <- pcaRes$pcvar

# we only want components that cpature 95% of the total variance or a little above.

# cutIdx is the index of the compoenent, within which the variance is just equal

# to or a little greater than 95% of the total.

cutIdx <- which(cumsum(pcvars)>0.95)

if(length(cutIdx)==0){

cutIdx <- componentsCount

}else{

cutIdx <- cutIdx[1]

}

# slice R and V to only that number of components.

R <- R[,1:cutIdx]

V <- V[,1:cutIdx]

# the difference between experiment mean and control mean.

meanvec <- rowMeans(expm) - rowMeans(ctrl)

# all the following steps calculate shrunkMats. Refer to the ChrDir paper for detail.

# ShrunkenMats are the covariance matrix that is placed as denominator

# in LDA formula. Notice the shrunkMats here is in the subspace of those components

# that capture about 95% of total variance.

Dd <- t(R)%\*%R/samplesCount

Dd <- diag(diag(Dd))

sigma <- mean(diag(Dd))

shrunkMats <- r\*Dd + sigma\*(1-r)\*diag(dim(R)[2])

# The LDA formula.

# V%\*%solve(shrunkMats)%\*%t(V) transforms the covariance matrix from the subspace to full space.

b <- V%\*%solve(shrunkMats)%\*%t(V)%\*%meanvec

# normlize b to unit vector

b <- b\*as.vector(sqrt(1/t(b)%\*%b))

# sort b to by its components' absolute value in decreasing order and get the

# sort index

sortRes <- sort(abs(b),decreasing=TRUE,index.return=TRUE)

# sort b by the sort index

bSorted <- as.matrix(b[sortRes$ix])

# sort genes by the sort index

genesSorted <- genes[sortRes$ix]

# assign genesSorted as the row names of bSorted

rownames(bSorted) <- genesSorted

# return bSorted

bSorted <- bSorted

}

* + What were the steps involved in performing the DE analysis?
    1. The code had to be modified to handle various types of data, rows, and input variables.
    2. The code above does the most important (in my opinion) and the most interesting analysis that I’d like to highlight. If we can explain this further in this section I’d like that.
  + Did you use any specific tools or libraries for the DE analysis? If so, which ones and why?
    1. #Load the necessary libraries
    2. library(GEOquery)
    3. library(DESeq2)
    4. setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
    5. # source("chdir.R")
    6. # source('nipals.R')
    7. library(readr)
    8. library(AnnotationDbi)
    9. library(org.Hs.eg.db)
    10. library(hgu133plus2.db)
    11. library(dplyr)
    12. library(org.Mm.eg.db)
    13. library(org.Rn.eg.db)
    14. main ones for DE analysis were limma for disease data and DESeq2 for drug data. All were then DE analyzed by chdir
  + What were the key findings from the DE analysis?

1. 1-Cosine Similarity
   * What is 1-cosine similarity, and why did you choose this metric?
     1. Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It calculates the cosine of the angle between the vectors, ranging from -1 to 1. A value of 1 indicates high similarity, while -1 indicates dissimilarity.
     2. In your code, you used 1-cosine similarity, which is the complement of cosine similarity. It ranges from 0 to 2, where 0 indicates high similarity and 2 indicates dissimilarity.
     3. The reason for choosing 1-cosine similarity is not explicitly stated in the code. However, it is commonly used in text mining and gene expression analysis to measure the similarity between gene expression profiles or document vectors.
   * How did you calculate the 1-cosine similarity in your study?
     1. In your code, you calculated the 1-cosine similarity using the cosine() function from the lsa library.
     2. You created a data matrix (data\_matrix) containing the gene expression values for both drug and disease data.
     3. You then calculated the cosine similarity between each pair of columns (representing drugs and diseases) in the data matrix using nested loops.
     4. The resulting cosine similarity values were stored in a symmetric matrix called cosine\_sim\_matrix.
     5. Finally, you subtracted the cosine similarity values from 1 to obtain the 1-cosine similarity.
   * What were the advantages of using 1-cosine similarity over other similarity measures?
     1. The code does not explicitly mention the advantages of using 1-cosine similarity over other measures. However, cosine similarity is known for its effectiveness in capturing the similarity between high-dimensional vectors, such as gene expression profiles or document vectors.
     2. Cosine similarity is insensitive to the magnitude of the vectors and focuses on the angle between them, making it suitable for comparing the relative patterns of gene expression or document content.
   * Did you encounter any limitations or challenges while using 1-cosine similarity?
     1. The code does not provide information about specific limitations or challenges encountered while using 1-cosine similarity.
     2. However, some common challenges with cosine similarity include its sensitivity to the normalization of the vectors and its inability to capture non-linear relationships between the vectors.
2. Results   
   a. Drug vs Disease
   * What were the main findings when comparing drugs and diseases?

Results

The cosine similarity matrix (subset\_matrix\_cleaned) provides valuable insights into the similarities between drug gene expression profiles and various kidney disease states. By analyzing this matrix, we can identify potential drug repurposing candidates and understand the underlying molecular mechanisms shared between drugs and diseases.

Cosine Similarity Matrix

The cosine similarity matrix reveals the degree of similarity between drug gene expression profiles and kidney disease profiles. Higher values indicate greater similarity between the drug and disease profiles. Notable findings from the cosine similarity matrix include:

Lisinopril (chdir\_Lisinopril\_GSE199437) exhibits the highest similarity across multiple kidney diseases, including MCD (1.1541379), Lupus Nephritis (1.1309973), Hypertensive Nephropathy (1.1228615), and FSGS (1.1151803). This suggests that Lisinopril may have potential therapeutic effects in these diseases.

JNJ39933673 (chdir\_JNJ39933673\_GSE199437) also shows high similarity with various kidney diseases, such as Diabetic Nephropathy (1.1312115), MCD (1.1194360), Hypertensive Nephropathy (1.1152950), and FSGS (1.0532960). This indicates that JNJ39933673 may be a promising drug candidate for further investigation in these disease contexts.

Ramipril (chdir\_ramipril\_GSE226353) demonstrates high similarity with Hypertensive Nephropathy (1.1183655) and Lupus Nephritis (1.0493342), suggesting its potential efficacy in treating these conditions.

Sparsentan (chdir\_Sparsentan\_GSE225447) exhibits high similarity with Diabetic Nephropathy (1.0882484), indicating its potential as a therapeutic option for this specific kidney disease.

Furosemide (chdir\_Furosemide\_GSE59913) shows high similarity with MGN (1.0486337), FSGS\_MCD (1.0491503), IgAN (1.0290862), and Hypertensive Nephropathy (1.0215214), suggesting its potential applicability in these kidney disorders.

Statistical Significance

To assess the statistical significance of the cosine similarity values, a permutation test was performed. The resulting p-values were categorized into different significance levels: p < 0.05 (\*), p < 0.01 (\*\*), and p < 0.001 (\*\*\*).

Some of the statistically significant similarities include:

Lisinopril (chdir\_Lisinopril\_GSE199437) shows highly significant similarity (p < 0.001, \*\*\*) with MCD, and significant similarity (p < 0.01, \*\*) with Lupus Nephritis, Hypertensive Nephropathy, and FSGS.

JNJ39933673 (chdir\_JNJ39933673\_GSE199437) exhibits significant similarity (p < 0.01, \*\*) with Diabetic Nephropathy and Hypertensive Nephropathy, and significant similarity (p < 0.05, \*) with FSGS\_MCD, FSGS, IgAN, Lupus Nephritis, and MCD.

Sparsentan (chdir\_Sparsentan\_GSE225447) shows highly significant similarity (p < 0.01, \*\*) with Diabetic Nephropathy and significant similarity (p < 0.05, \*) with IgAN.

Interpretation and Implications

The cosine similarity matrix provides a comprehensive view of the similarities between drug gene expression profiles and kidney disease profiles. The results highlight several drugs, such as Lisinopril, JNJ39933673, Ramipril, Sparsentan, and Furosemide, that exhibit significant similarity with multiple kidney diseases. These drugs may have potential for repurposing and warrant further investigation in preclinical and clinical studies.

The identification of statistically significant similarities between specific drugs and diseases can guide future research efforts to elucidate the shared molecular pathways and mechanisms of action. This knowledge can aid in the development of targeted therapies and personalized treatment strategies for kidney diseases.

However, it is important to note that these findings are based on gene expression data and require validation through additional experimental and clinical studies. The cosine similarity matrix serves as a valuable starting point for generating hypotheses and prioritizing drug candidates for further exploration in the context of kidney diseases.

b. Drug vs Drug

* + Results
  + The cosine similarity matrix (drug\_matrix\_cleaned) provides valuable insights into the similarities between gene expression profiles of different drugs tested on kidney tissue. By analyzing this matrix along with the corresponding significance matrix, we can identify drugs with similar effects on kidney gene expression and potential drug repurposing candidates.
  + Cosine Similarity Matrix
  + The cosine similarity matrix reveals the degree of similarity between drug gene expression profiles. Higher values indicate greater similarity between the drugs' effects on kidney gene expression. Notable findings from the cosine similarity matrix include:
  + Ramipril (chdir\_ramipril\_GSE226353) and Sparsentan (chdir\_Sparsentan\_GSE225447) exhibit the highest similarity among all drug pairs, with a cosine similarity value of 1.5453496. This suggests that these two drugs may have similar effects on kidney gene expression and could potentially be repurposed for similar indications.
  + JNJ39933673 (chdir\_JNJ39933673\_GSE199437) shows high similarity with Ramipril (chdir\_ramipril\_GSE226353) (1.4136024) and Candesartan (chdir\_Candesartan\_GSE59913) (1.1942767), indicating that these drugs may share similar mechanisms of action in the kidney.
  + Lisinopril (chdir\_Lisinopril\_GSE199437) demonstrates high similarity with Ramipril (chdir\_ramipril\_GSE226353) (1.2530703), suggesting that these ACE inhibitors may have comparable effects on kidney gene expression.
  + Spironolactone (chdir\_Spironolactone\_GSE59913) and Finerenone (chdir\_Fineronone\_GSE183841) show high similarity (1.2222069), indicating potential similarities in their effects on kidney gene expression, which is consistent with their shared mechanism of action as mineralocorticoid receptor antagonists.
  + Bicarbonate (chdir\_Bicarbonate\_GSE200638) exhibits high similarity with several drugs, including Sparsentan (chdir\_Sparsentan\_GSE225447) (1.2379868), Spironolactone (chdir\_Spironolactone\_GSE59913) (1.1547340), and Prednisone (chdir\_Prednisone\_GSE59913 and chdir\_Prednisone\_GSE153021) (1.1082566 and 1.1138565, respectively). This suggests that bicarbonate treatment may have overlapping effects on kidney gene expression with these drugs.
  + Statistical Significance
  + The significance matrix provides information about the statistical significance of the cosine similarity values. Asterisks (\*, \*\*, \*\*\*) indicate the level of significance, with more asterisks representing higher significance.
  + Some of the statistically significant similarities include:
  + Ramipril (chdir\_ramipril\_GSE226353) and Sparsentan (chdir\_Sparsentan\_GSE225447) show highly significant similarity (p < 0.001, \*\*\*).
  + JNJ39933673 (chdir\_JNJ39933673\_GSE199437) exhibits significant similarity with Ramipril (chdir\_ramipril\_GSE226353) (p < 0.01, \*\*) and Candesartan (chdir\_Candesartan\_GSE59913) (p < 0.05, \*).
  + Spironolactone (chdir\_Spironolactone\_GSE59913) and Finerenone (chdir\_Finerenone\_GSE183841) have significant similarity (p < 0.05, \*).
  + Interpretation and Implications
  + The cosine similarity matrix and significance matrix provide valuable information about the similarities between drug gene expression profiles in kidney tissue. The results highlight several drugs with significant similarities, such as Ramipril and Sparsentan, JNJ39933673 and Ramipril/Candesartan, Lisinopril and Ramipril, and Spironolactone and Finerenone. These findings suggest that these drugs may share similar mechanisms of action or have overlapping effects on kidney gene expression.
  + The identification of significant similarities between drugs can guide further research into potential drug repurposing opportunities. Drugs with high similarity scores and statistical significance could be prioritized for preclinical and clinical studies to explore their therapeutic potential in kidney-related disorders.
  + Moreover, understanding the similarities between drug gene expression profiles can provide insights into the underlying molecular mechanisms and pathways affected by these drugs in the kidney. This knowledge can aid in the development of targeted therapies and personalized treatment strategies for kidney diseases.
  + However, it is important to note that these findings are based on gene expression data from kidney tissue and may not fully capture the complexities of drug effects in vivo. Additional experimental and clinical validation is necessary to confirm the therapeutic potential and safety of repurposing these drugs for kidney-related indications.

1. Data Summary a. Data Availability
   * Where can readers access the datasets used in your study?
   * Are there any restrictions or limitations on data availability?
   * Did you create any new datasets or derived data products during your study? If so, how can others access them?
2. References
   * Did you cite all the relevant literature and sources used in your study?
   * Are the references properly formatted according to the required citation style?

Additional questions to consider:

* What motivated you to undertake this independent study?
* What prior research or existing knowledge did you build upon?
* Did you face any challenges or limitations during the study? How did you address them?
* What are the potential future directions or extensions of your work?
* How does your study contribute to the broader field of research?