What this presentation will do?

In the presentation, I will tell you about...

1. what is my task.
2. what did I do.
3. what difficulties I met.
4. what the code does.
5. what is the implementation.
6. how I read `*.gmt` into R.
7. what am I going to do.

My task

Optimize the code to take in several pathways from `MSigDB`.

- Read in GMT files of pathways ✅
- Recompute $D_{pathway}$ for each pathway of interest ✅
- Re-execute permutation testing & z-score calculation
- Generate top drug hits against query drugs
- Show communities arising per pathway (see `DNF` herokuapp)

What did I do?

1. I setup the development environment
2. I read and try to understand the following codes:
    1. `DeenaGendoo_Generate_MVA_DNF.R` ✅
    2. `DeenaGendoo_PermutationTestAndFiltering.R`
3. I read `KEGG.gmt` into R.

What difficulties I met?

I am on a M1 BigSur

- Compatibility issue.

> *Do not buy M1 until your stuffs are ready for M1.* *Chris, 2021*

What I understood from the code

- 🗒️ **Code**: `DeenaGendoo_Generate_MVA_DNF.R`
- 🎯 **Goal**: To have an network that allows drugs comparison for a specific biochemical pathway.
- 🛠️ **Methodology**: Create an *integrated network* for the *mevalonate pathway* that shows all the relationships between drugs and themselves by constructing different *network layers* from the given `PrePros.RData`.
- 📁 **Output**: Save the *integrated network* and constructed *layers* to a `MVA_DNF.RData`

Terminologies 

What is mevalonate pathway?

- *mevalonate pathway* (`mva`) is a biochemical pathway.
- `MVA_genes.csv` are the genes involved in the `mva`

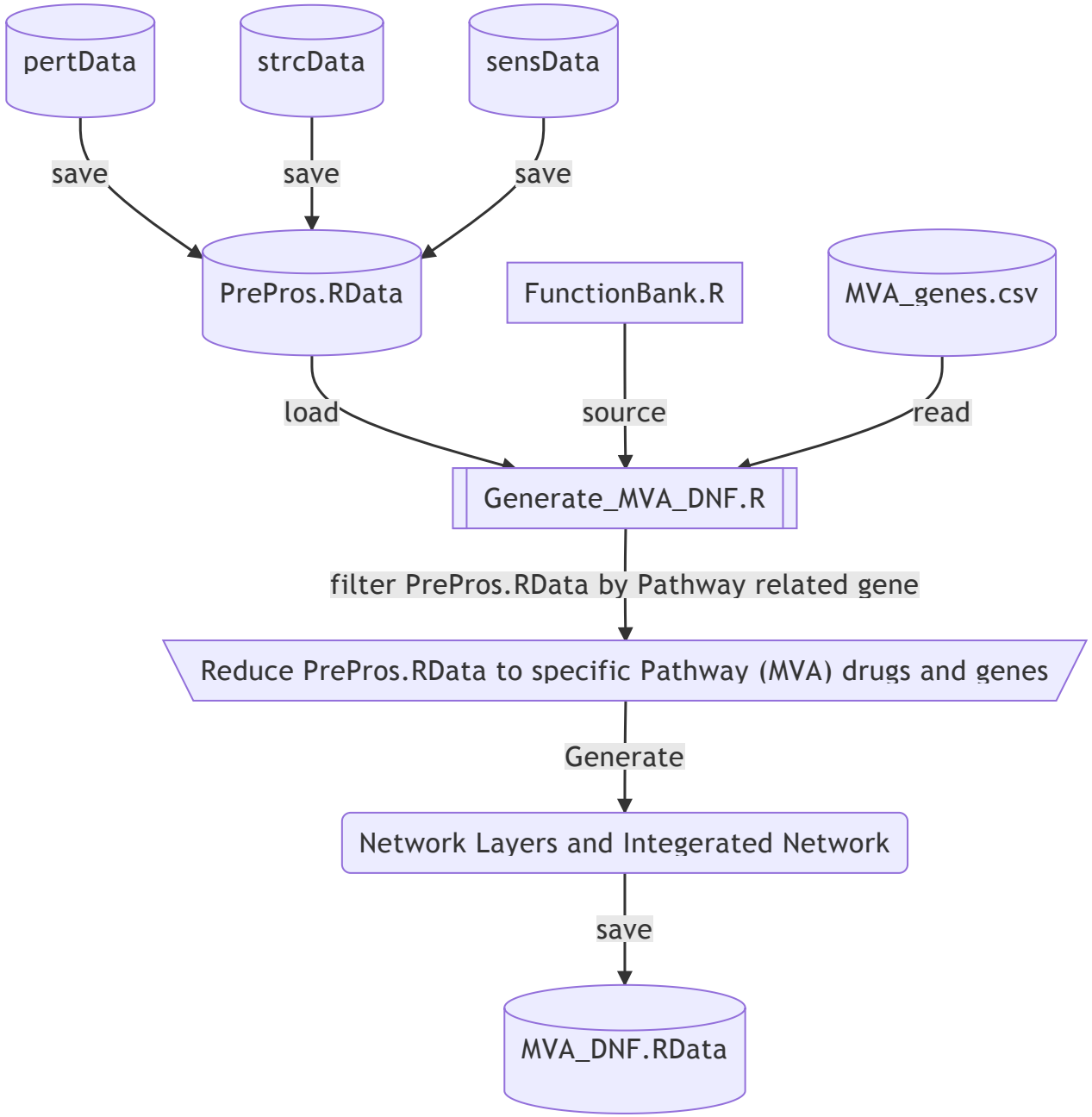
What are network layers?

- Essentially, they are all correlation matrices of drugs, i.e. `drug-drug correlation matrix`.
- They describe the drugs similarity with respect to different profiles i.e. data set. which are the followings:
    - **Perturbation profile**: Perturbation between **Drug** and **Gene**
    - **Sensitivity profile**: Sensitivity between **Drug** and **Cell line**
    - **Drug Structure profile**: Structure between **Drug** and **Drug**
- They are converted to an *Affinity Matrix* for the networks integration eventually.

What is integrated network?

- It is the main goal.
- It is, still, a `drug-drug correlation matrix`.
- It is integrated by the following 3 network layers *Affinity Matrix*:
    1. Drug Perturbation: `pertAffMat`
    2. Drug Sensitivity: `sensAffMat`
    3. Drug Structure: `strcAffMat`
- It uses *similarity network fusion (SNF)* method to integrate different networks.
- It shows all the relationships between drugs and themselves

Overview: `DeenaGendoo_Generate_MVA_DNF.R`


pertData
strcData
sensData
save
save
save
PrePros.RData
FunctionBank.R
MVA_genes.csv
load
source
read
Generate_MVA_DNF.R
filter PrePros.RData by Pathway related gene
Reduce PrePros.RData to specific Pathway (MVA) drugs and genes
Generate
Network Layers and Integerated Network
save
MVA_DNF.RData

Data: `PrePros.RData`

obj	`class(.)`	shape	Description	source
`pertData`	`matrix, array`	row: 978, col: 238	**Drug-Gene** Perturbation	LINCS-L1000
`sensData`	`matrix, array`	row: 60, col:238	**Drug-Cell line** Sensitivity	NCI-60
`strcData`	`list, fingerprint object`	length: 238	**Drugs** Structure	

`FunctionBank.R`

It contains functions that ...

- constuct the respective network layers
    - `constPerturbationLayer(pertDat)`
    - `constSensitivityLayer(sensDat)`
    - `constStructureLayer(strcDat)`
- intgerate the network layers
    - `integrateStrctSensPert(sensAff, strcAff, pertAff)`

`MVA_gene.csv`

- It contains the genes that involve in the `mva` pathway.

⚠️ In the future development, it will be replaced by other `*.gmt` such that we can build networks for other pathways

Implementation 📌

`DeenaGendoo_Generate_MVA_DNF.R` does the following:

1. Load packages
2. ⭐ Read and load the target biochemical pathway, i.e. read `MVA_genes.csv` to `mva` object.
3. Extract the gene names that involved in `mav`, i.e. `mvagenes`.
4. Reduce the gene in `pertData` to `mvagenes`.
5. Reduce the drugs in all data sets to that remains in `pertData`, i.e keep the drug that related to `mva`.
6. ⭐ Construct network layers for Structure, Sensitivity and Perturbation: `strcAffMat`, `sensAffMat`, `pertAffMat`.
7. ⭐ Integrate all the network layers to a single integrated network: `integrtStrctSensPert`
8. Save all layers and the integrated network to a `.RData`.

Gist ⭐

- The main things happens at 2️⃣, 6️⃣, and 7️⃣
- 2️⃣ reads in the specific biochemcal pathway from `MVA_genes.csv`
  - It will be replaced by `*.GMT` files later for other pathways.
  - I will obtain `*.GMT` from **MSigDB**.
- 6️⃣ contructs the networks layers
- 7️⃣ integerates the networks layers that contructed in 6️⃣.

Network layers and integeration

Practically, It is the following snippet:

```
strcAffMat <- constStructureLayer(strcData)
sensAffMat <- constSensitivityLayer(sensData)
pertAffMat <- constPerturbationLayer(pertData)
integrtStrctSensPert <- integrateStrctSensPert(sensAffMat, strcAffMat, pertAffMat)
```

Reading `*.gmt`

- I use GSA 

```
Description:

     Read in a gene set collection from a .gmt file

Usage:

     GSA.read.gmt(filename)

Value:

A list with components

     genesets: List of gene names (identifiers) in each gene set,
     geneset.names: Vector of gene set names (identifiers),
     geneset.descriptions: Vector of gene set descriptions
```

Let's take `kegg.gmt` as an example.

`kegg.gmt` description

- `c2.cp.kegg.v7.4.symbols.gmt`
- Canonical Pathways gene sets derived from the KEGG pathway database.
- 186 gene set i.e pathways.

Reading `kegg.gmt` into R in action

```
r$> library(GSA)
r$> kegg <- GSA.read.gmt("Data/GMT/c2.cp.kegg.v7.4.symbols.gmt")

r$> length(kegg$geneset.names)
[1] 186

r$> kegg$geneset.names[1]
[1] "KEGG_N_GLYCAN_BIOSYNTHESIS"
```

kegg$geneset.names[1] genesets

```
r$> kegg$genesets[[1]]
[1] "ALG13"    "DOLPP1"   "RPN1"     "ALG14"    "MAN1B1"   "ALG3"     "B4GALT1" "MGAT5"
[9] "RPN2"     "STT3A"    "MGAT3"    "DAD1"     "MGAT2"    "ALG12"    "TUSC3"    "MAN1C1"
[17] "DPM2"     "DPM1"     "GANAB"    "ALG1"     "MGAT4A"   "ALG10B"   "STT3B"    "MAN1A2"
[25] "ALG10"    "ALG11"    "ALG8"     "ALG2"     "DPAGT1"   "RFT1"     "DPM3"     "DDOST"
[33] "MGAT4B"   "ALG6"     "MAN2A2"   "MAN1A1"   "MAN2A1"   "ST6GAL1" "B4GALT3" "ALG5"
[41] "B4GALT2" "MGAT5B"   "ALG9"     "MOGS"     "FUT8"     "MGAT1"

r$> kegg$genesets[[1]][1]
[1] "ALG13"
```

`kegg$geneset.names[1]` description

```
r$> kegg$geneset.descriptions[[1]]
[1] "http://www.gsea-msigdb.org/gsea/msigdb/cards/KEGG_N_GLYCAN_BIOSYNTHESIS"
```

Replacing `mvagenes`

Notice that `mvagenes` is a list of `character`, see below.

```
r$> mva <- read.csv("Data/MVA_genes.csv")

r$> class(mva)
[1] "data.frame"

r$> mvagenes <- as.character(sort(mva$Gene.name))

r$> class(mvagenes)
[1] "character"

r$> mvagenes
 [1] "ACAT2"   "ACLY"    "ACSS2"   "DPAGT1" "FDFT1"
 [6] "FDPS"    "GGPS1"   "HMGCR"   "HMGCS1" "INSIG1"
[11] "LDLR"    "MVD"     "MVK"     "SREBP2"

r$> mvagenes[1]
[1] "ACAT2"
```

`mvagenes` equivalent `kegg$genesets`

- to replace `mvagenes` by the equivalent `kegg$genesets` , we should use `kegg$genesets[[i]]` but *NOT* `kegg$genesets[i]`, where `1 <= i <= 186`
- see difference below:

```
r$> class(kegg$genesets[[1]])
[1] "character"

r$> class(kegg$genesets[1])
[1] "list"
```

what am I going to do?

I will...

- Try to read the rest of `*.gmt` files for the pathways
  - `c2.cp.kegg.v7.4.symbols.gmt`
  - `c2.cp.reactome.v7.4.symbols.gmt`
- Try to replicate the DNF for pathways in `*.gmt`
- continue reading the `DeenaGendoo_PermutationTestAndFiltering.R`

Thank you for your attention