Stat4DS / Homework 03

Pierpaolo Brutti

Due whenever you want before February 20

General Instructions

I expect you to upload your solutions on Moodle as a **single running** `R Markdown` file (`.rmd`) + its `html` output, **named with your surnames**. Alternatively, a `zip`-file with all the material inside will be fine too.

R Markdown Test

To be sure that everything is working fine, start `RStudio` and create an empty project called `HW3`. Now open a new `R Markdown` file (`File > New File > R Markdown...`); set the output to `HTML mode`, press `OK` and then click on `Knit HTML`. This should produce a web page with the knitting procedure executing the default code blocks. You can now start editing this file.

Who let the DAGs out?

Remember DAGs? Good. It's now time to learn-how-to-learn their topology (at least in the Gaussian case) and then put them to work in a biological setup.

Our statistical recipe needs a lot of ingredients, namely:

1. The basics of the *Likelihood Ratio Test* (LRT) method.
2. The concept and ideas behind *Universal Inference*.
3. The notion of *Gaussian DAGs* and their (constrained) likelihood functions
4. The incarnation of LRT for testing directed connections in Gaussian DAGs

. . . I guess, we better start. . .

Ingredient (A): The Likelihood Ratio Test

The Wald test is useful for testing a scalar parameter. The Likelihood Ratio Test (LRT) is more general and can be used for testing a vector–valued parameter. More specifically:

The Likelihood Ratio Test

Within a parametric model $\mathcal{F} = \{f_{\boldsymbol{\theta}}(\cdot) : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\}$, consider testing

$$H_0 : \boldsymbol{\theta} \in \Theta_0 \quad \text{VS} \quad H_1 : \boldsymbol{\theta} \notin \Theta_0$$

Given an IID sample $\mathcal{D}_n = \{X_1, \ldots, X_n\} \sim f_{\boldsymbol{\theta}}(\cdot)$ with the associated likelihood function $\mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n) = \prod_i f_{\boldsymbol{\theta}}(X_i)$, the likelihood ratio test rejects the null if $U_n > c$ for some suitable *critical level* c, with

$$U_n = \frac{\sup_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n)}{\sup_{\boldsymbol{\theta} \in \Theta_0} \mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n)} = \frac{\mathcal{L}(\widehat{\boldsymbol{\theta}} \,|\, \mathcal{D}_n)}{\mathcal{L}(\widehat{\boldsymbol{\theta}}_0 \,|\, \mathcal{D}_n)},$$

where $\widehat{\boldsymbol{\theta}}_0$ denotes the constrained to Θ_0 MLE (i.e. assuming H_0 is true), whereas $\widehat{\boldsymbol{\theta}}$ denotes the unconstrained MLE.

Remarks:

- Replacing Θ_0^c with Θ in the denominator has little effect on the test statistic and the unconstrained version simplifies the theoretical properties of the test statistics.
- The likelihood ratio test is most useful when Θ_0 consists of all parameter values $\boldsymbol{\theta}$ such that some coordinates of $\boldsymbol{\theta}$ are fixed at particular values.

Let's now look at a famous example on testing for the mean of a **Normal population**: one of the few cases where we have **exact**, **finite sample**, results.

Example: Student's t–test

Let $\mathcal{D}_n = \{X_1, \ldots, X_n\}$ be IID from a $\mathrm{N}(\mu, \sigma^2)$ and we want to test

$$H_0 : \mu = \mu_0 \quad \text{vs} \quad H_1 : \mu \neq \mu_0 \quad \rightsquigarrow \quad U_n = \frac{\mathcal{L}(\widehat{\mu}, \widehat{\sigma} \,|\, \mathcal{D}_n)}{\mathcal{L}(\mu_0, \widehat{\sigma}_0 \,|\, \mathcal{D}_n)},$$

where $\widehat{\sigma}_0$ maximizes the likelihood under the null, that is, subject to $\mu = \mu_0$.
After some simple but tedious algebra, it can be shown that $U_n > c \Leftrightarrow T_n > c'$ where

$$T_n = \frac{\bar{X}_n - \mu_0}{\widehat{\sigma}/\sqrt{n}} \overset{\text{under } H_0}{\sim} t_{n-1} \quad \text{with} \quad \widehat{\sigma}^2 = \frac{1}{n}\sum_i (X_i - \bar{X}_n)^2.$$

So the final two–sided test on the mean μ of a **Normal population** is:

$$\text{Reject } H_0 \text{ if } |T_n| > t_{n-1,\alpha/2} \qquad (\text{Student's } t\text{–test}).$$

Similarly to the Wald test, in more general situations where we are dealing with non-Gaussian (but still regular!) populations, all we can do to appropriately tune the critical value c in order to control the type-I error probability of our LRT, is to appeal to some suitable, broadly applicable, asymptotic result.

More specifically, here's two classics:

Asymptotic approximation / scalar

Consider testing $H_0 : \theta = \theta_0$ versus $H_1 : \theta \neq \theta_0$ where $\theta \in \mathbb{R}$.
Then, under H_0 (+ regularity conditions on the population model $\mathcal{F}$),

$$T_n = 2 \log U_n \xrightarrow{d} \chi^2_1.$$

Hence an asymptotic level α test is: Reject H_0 when $T_n > \chi^2_{1,\alpha}$.

Asymptotic approximation / vector

Consider testing a null $H_0 : \boldsymbol{\theta} \in \Theta_0 \subseteq \mathbb{R}^p$ where we are fixing some parameters.
Then, under H_0 (+ regularity conditions on the population model $\mathcal{F}$),

$$T_n = 2 \log U_n \xrightarrow{d} \chi^2_\nu \quad \text{where} \quad \nu = \dim(\Theta) - \dim(\Theta_0).$$

Hence an asymptotic level α test is: Reject H_0 when $T_n > \chi^2_{\nu,\alpha}$.

Ingredient (B): Universal Inference

As we all should know at this point, in classical frequentist statistics, confidence sets and tests are often obtained by starting from **asymptotically Gaussian estimators** or other large sample results.
As a consequence, their validity relies on large sample asymptotic theory and requires that the statistical model satisfy certain regularity conditions. When these conditions do **not** hold, or the sample is not large "enough", there is **no general method** for statistical inference, and these settings are typically considered in an ad-hoc manner.

Recently a new, universal method simply based on **sample splitting** has been introduced which yields tests and confidence sets for **any** statistical model (regular or not) and comes also with **finite-sample** guarantees.

Focusing on **hypothesis testing**, historically speaking **sample splitting** was first analysed computationally and theoretically in a 1975-4-pages-long paper by sir David Cox, one of the greatest statisticians of all times, and then further discussed in his 1977 review (2, Section 3.2), where he describes the method as well known and refers to an American Statistician paper with a wide-ranging discussion of *"snooping"*, *"fishing"*, and *"hunting"* in data analysis.
Honestly? An easy read way more relevant now than then!

Let's now describe this idea in the context of LRT:

Universal Hypothesis Test

Let $\mathcal{D}_{2n} = \{X_1, \ldots, X_{2n}\}$ be an IID sample from a population model $\mathcal{F}$ having density $f_{\boldsymbol{\theta}}(\cdot)$ and consider testing

$$H_0 : \boldsymbol{\theta} \in \Theta_0 \quad \text{VS} \quad H_1 : \boldsymbol{\theta} \notin \Theta_0.$$

To this end, (randomly) **split** the data $\mathcal{D}_{2n}$ in two groups having the same size n (just to simplify the notation), and build the two corresponding likelihood functions:

$$\mathcal{D}_{2n} \rightsquigarrow \left\{\mathcal{D}_n^{\mathtt{Tr}}, \mathcal{D}_n^{\mathtt{Te}}\right\} \rightsquigarrow \left\{\mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n^{\mathtt{Tr}}) = \textstyle\prod_{i \in \mathcal{D}_n^{\mathtt{Tr}}} f_{\boldsymbol{\theta}}(X_i), \quad \mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n^{\mathtt{Te}}) = \prod_{i \in \mathcal{D}_n^{\mathtt{Te}}} f_{\boldsymbol{\theta}}(X_i)\right\}$$

Consider now the following two MLE's for $\boldsymbol{\theta}$

$$\underbrace{\widehat{\boldsymbol{\theta}}_0^{\mathtt{Tr}} = \underset{\boldsymbol{\theta} \in \Theta_0}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n^{\mathtt{Tr}})}_{H_0\text{-constrained estimator based on the Training Data}} \qquad \underbrace{\widehat{\boldsymbol{\theta}}^{\mathtt{Te}} = \underset{\text{All } \boldsymbol{\theta}}{\operatorname{argmax}}\, \mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n^{\mathtt{Te}})}_{\text{Unconstrained estimator based on the Test Data}}$$

At this point we are ready to define the following two **universal test statistics**:

$$\underbrace{U_n = \frac{\mathcal{L}\big(\widehat{\boldsymbol{\theta}}^{\mathtt{Te}} \,|\, \mathcal{D}_n^{\mathtt{Tr}}\big)}{\mathcal{L}\big(\widehat{\boldsymbol{\theta}}_0^{\mathtt{Tr}} \,|\, \mathcal{D}_n^{\mathtt{Tr}}\big)}}_{\text{Split Likelihood Ratio}} \qquad \text{and} \qquad \underbrace{W_n = \frac{U_n + U_n^{\mathtt{swap}}}{2}}_{\text{Cross-Fit Likelihood Ratio}} \tag{1}$$

where $U_n^{\mathtt{swap}}$ is the same as U_n after *swapping* the roles of $\mathcal{D}_n^{\mathtt{Tr}}$ and $\mathcal{D}_n^{\mathtt{Te}}$.
Based on U_n and W_n, we get the following two **univeral testing procedures**:

$$\underbrace{\text{Reject } H_0 \text{ if } U_n > \tfrac{1}{\alpha},}_{\text{Split LRT}} \qquad \text{and} \qquad \underbrace{\text{Reject } H_0 \text{ if } W_n > \tfrac{1}{\alpha}}_{\text{Cross-Fit LRT}} \tag{2}$$

By simply using **Markov inequality** and under **no assumptions** on the population model $\mathcal{F}$, in *Theorem 3* the Authors show that in **finite sample** the Split and Cross-Fit LRTs control the type-I error probability at level α.

Ingredient (C): Gaussian DAGs and their (constrained) Likelihood

We (should) know that we use DAG models to encode the joint distribution $f(\boldsymbol{x})$ of a random vector $\boldsymbol{X} = [X_1, \ldots, X_p]^{\mathsf{T}} \in \mathbb{R}^p$: the nodes and the **directed** edges represent, respectively, the variables and the parent-child dependence relations between any two variables. We also know that the joint $f(\boldsymbol{x})$ is Markov w.r.t. a graph $\mathcal{G}$ if it admits the following factorization

$$f(\boldsymbol{x}) = \prod_{j=1}^{p} f\big(x_j \,|\, \mathrm{pa}(x_j)\big),$$

where $\mathrm{pa}(x_j)$ denotes the set of variables with an arrow towards X_j in the (directed) graph $\mathcal{G}$.

Our main goal here, is to lay down a strategy to infer the pairwise relation imposed by the (local) Markov dependence. To get started, first of all we restrict the scope of our analysis focusing on a Gaussian random vector $\boldsymbol{X} \sim \mathrm{N}_p$.

Under this assumption, we can capture the directional effects induced by directed edges by using a **linear structural equation model**[1]

$$X_j = \sum_{k\,:\,k \neq j} \mathbb{A}[j,k] \cdot X_k + \epsilon_j \quad \text{where} \quad \epsilon_j \overset{\text{IID}}{\sim} \mathrm{N}_1(0, \sigma^2), \tag{3}$$

and $\mathbb{A}$ is a $(p \times p)$ adjacency matrix in which a nonzero entry $\mathbb{A}[j,k]$ in position (j,k) corresponds to a directed edge from parent node k to child node j, with its value indicating the strength of the relation; $\mathbb{A}[j,k] = 0$ when $k \notin \mathrm{pa}(X_j)$.

Learning Goal

Given a random sample $\mathcal{D}_n = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\} \overset{\text{IID}}{\sim} f(\cdot)$, where $\boldsymbol{X}_i = [X_{i,1}, \ldots, X_{i,p}]^{\mathsf{T}} \in \mathbb{R}^p$, infer the adjacency matrix $\mathbb{A}$ **subject to** the requirements that $\mathbb{A}$ defines a directed **acyclic** graph.

Denoting by $\boldsymbol{\theta} = (\mathbb{A}, \sigma^2)$ the parameters of our model, to approach the problem from a maximum likelihood perspective, we need to write down:

1. The (log-)likelihood function $\ell_n(\boldsymbol{\theta}) = \ln \mathcal{L}(\boldsymbol{\theta} \,|\, \mathcal{D}_n)$.
2. How to introduce *acyclicity* constraints to drive the learning process.

[1] The homoscedasticity of the error $\{\epsilon_j\}_j$ is not required but useful to induce *identifiability* and avoid technicalities regarding equivalence classes. In addition, individual means μ_j could be incorporated by adding intercepts to Equation 3. For simplicity, in what follows we set the means to zero.

By collecting all the available data in an $(n \times p)$ matrix $\mathbb{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_i \cdots \boldsymbol{x}_n]^{\intercal}$, and because of the Gaussian nature of the structural equation model in Equation 3, the log-likelihood is readily available:

$$\ell_n(\mathbb{A}, \sigma^2) = -\sum_{j=1}^{p} \left(\frac{1}{2\sigma^2} \sum_{i=1}^{n} \Big(\mathbb{X}[i,j] - \textstyle\sum_{k\,:\,k\neq j} \mathbb{A}[j,k] \cdot \mathbb{X}[j,k] \Big)^2 + \frac{n}{2} \ln \sigma^2 \right). \tag{4}$$

The *acyclicity* constraints are all but trivial. Fortunately, the hard work has already been done by Yuan et al. in 2019[2]. Based on their *Theorem 1*, we know that $\mathbb{A}$ satisfies the acyclicity constraints if and only if there exist a $(p \times p)$ matrix Λ s.t.:

$$\Lambda[r,k] + \mathbb{1}(j \neq k) - \Lambda[j,k] \geqslant \mathbb{1}\big(\mathbb{A}[r,j] \neq 0\big) \quad \text{for all} \quad r,j,k \in \{1,\ldots,p\} \text{ and } r \neq j. \tag{5}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. Although the matrix Λ may not be unique, Equation 5 has three clear pros:

1. It reduces the original super-exponentially many constraints on $\mathbb{A}$ to $p^3 - p^2$ active constraints on $(\mathbb{A}, \Lambda)$.
2. Code acyclicity in a simple way: each constraint involves only one parameter in $\mathbb{A}$ and is linear in $\Lambda[j,k]$ and $\mathbb{1}\big(\mathbb{A}[r,j] \neq 0\big)$.
3. The non-convexity of the constraints in Equation 5 due to the presence of the indicator $\mathbb{1}\big(z \neq 0\big)$ can be easily handled by replacing it with its computational surrogate: the *truncated L_1-function*: $J_\tau(z) = \min(|z|/\tau, 1)$, a piecewise linear function that converges to the indicator as $\tau \downarrow 0$. This yields the following approximated set of acyciclity constraints:

$$\Lambda[r,k] + \mathbb{1}(j \neq k) - \Lambda[j,k] \geqslant J_\tau\big(\mathbb{A}[r,j]\big) \quad \text{for all} \quad r,j,k \in \{1,\ldots,p\} \text{ and } r \neq j. \tag{6}$$

Now, before putting everything together in a test statistics, one last addition is the introduction of a **sparsity/complexity constraint** on $\mathbb{A}$, very useful to handle high-dimensional setups where $p >> n$:

$$\sum_{r,j\,:\,r\neq j} \mathbb{1}\big(\mathbb{A}[r,j] \neq 0\big) \leqslant \kappa \quad \rightsquigarrow \quad \sum_{r,j\,:\,r\neq j} J_\tau\big(\mathbb{A}[r,j]\big) \leqslant \kappa \tag{7}$$

Here $\kappa > 0$ is a tuning parameter to be selected possibly via a suitable grid-search.

Learning Problem

Squeezing together Equations 4, Equation 6 and Equation 7, the **directed acyclic graph learning problem** can be reformulated from a *constrained* maximum likelihood perspective as follow

$$\begin{aligned} & & \big(\widehat{\mathbb{A}}, \widehat{\sigma}^2\big) &= \underset{(\mathbb{A},\sigma^2,\Lambda)}{\operatorname{argmax}} \ \ell_n(\mathbb{A}, \sigma^2) \\ & \text{subject to} & & \sum_{r,j\,:\,r\neq j} J_\tau\big(\mathbb{A}[r,j]\big) \leqslant \kappa \\ & & & \Lambda[r,k] + \mathbb{1}(j \neq k) - \Lambda[j,k] \geqslant J_\tau\big(\mathbb{A}[r,j]\big) \quad \text{for all} \quad r,j,k \in \{1,\ldots,p\} \text{ and } r \neq j \end{aligned} \tag{8}$$

Ingredient (D): Constrained Likelihood Ratio Test

Now that we know how to fit our Gaussian model enforcing acyclicity (and sparsity!), we are finally in position to build some dedicated LRT to check the presence of some specific *directed* connection of interest. Along the lines of Li et al. (2019)[3], we have two possible types of tests concerning directional pairwise relations, encoded by an adjacency matrix $\mathbb{A}$. More specifically:

1. **Test of Graph Linkages**
   Let F be an index set where an index $(j,k) \in F$ represents a directed connection. We are interested in testing:

$$H_0 : \mathbb{A}[j,k] = 0 \text{ for } \underline{\textbf{all}}\ (j,k) \in F \quad \text{vs} \quad H_1 : \text{not } H_0 \qquad \rightsquigarrow \qquad \underset{\text{Constrained Likelihood Ratio Statistics}}{U_n = \frac{\mathcal{L}\big(\widehat{\mathbb{A}}, \widehat{\sigma}^2 \,\big|\, \mathcal{D}_n\big)}{\mathcal{L}\big(\widehat{\mathbb{A}}_0, \widehat{\sigma}_0^2 \,\big|\, \mathcal{D}_n\big)}} \tag{9}$$

   where $\big(\widehat{\mathbb{A}}, \widehat{\sigma}^2\big)$ are the unconstrained MLEs solving the optimization problem in Equation 8, whereas $\big(\widehat{\mathbb{A}}_0, \widehat{\sigma}_0^2\big)$ are the H_0-constrained MLEs solving the following H_0-augmented optimization problem

$$\begin{aligned} & & \big(\widehat{\mathbb{A}}, \widehat{\sigma}^2\big) &= \underset{(\mathbb{A},\sigma^2,\Lambda)}{\operatorname{argmax}} \ \ell_n(\mathbb{A}, \sigma^2) \\ & \text{subject to} & & \mathbb{A}[F] = 0 \\ & & & \textstyle\sum_{r,j\,:\,r\neq j} J_\tau\big(\mathbb{A}[r,j]\big) \leqslant \kappa \\ & & & \Lambda[r,k] + \mathbb{1}(j \neq k) - \Lambda[j,k] \geqslant J_\tau\big(\mathbb{A}[r,j]\big) \quad \text{for all} \quad r,j,k \in \{1,\ldots,p\} \text{ and } r \neq j \end{aligned} \tag{10}$$

[2] An implementation of their method is available at: https://github.com/ciryiping/TLPDAG.

[3] An implementation of their method is available at: https://github.com/chunlinli/clrdag. Notice that our sparsity parameter κ is `mu` in the main function `MLEdag()`. To install this package on `OSX`, you need to *"brew"* `gcc` and then make it visible to `R`. Please read this and/or contact us.

2. **Test of Directed Pathway**
   A directed pathway is specified by an index set F of size $|F|$ where a common segment is shared by any two consecutive indices, like $F = \big\{(j_1, j_2), (j_2, j_3), \ldots, (j_{|F|-1}, j_{|F|})\big\}$. We are interested in testing:

$$H_0 : \mathbb{A}[j,k] = 0 \text{ for } \underline{\textbf{some}}\ (j,k) \in F \quad \text{vs} \quad H_1 : \mathbb{A}[j,k] \neq 0 \text{ for } \underline{\textbf{all}}\ (j,k) \in F \ \rightsquigarrow\ \underbrace{U_n = \frac{\mathcal{L}\big(\widehat{\mathbb{A}}, \widehat{\sigma}^2 \,\big|\, \mathcal{D}_n\big)}{\max_{k=1}^{|F|} \mathcal{L}\big(\widehat{\mathbb{A}}_{0,k}, \widehat{\sigma}^2_{0,k} \,\big|\, \mathcal{D}_n\big)}}_{\text{Constrained Likelihood Ratio Statistics}} \qquad (11)$$

where $(\widehat{\mathbb{A}}, \widehat{\sigma}^2)$ are the unconstrained MLEs solving the optimization problem in Equation 8, whereas $(\widehat{\mathbb{A}}_{0,k}, \widehat{\sigma}^2_{0,k})$ are the H_0-constrained MLEs solving, for each $k \in \{1, \ldots, |F|\}$, the following H_0-augmented optimization problem:

$$\begin{aligned} &\big(\widehat{\mathbb{A}}, \widehat{\sigma}^2\big) = \underset{(\mathbb{A}, \sigma^2, \Lambda)}{\text{argmax}}\ \ell_n(\mathbb{A}, \sigma^2) \\ \text{subject to} \quad & \mathbb{A}[j_k, j_{k+1}] = 0 \text{ for } (j_k, j_{k+1}) \in F \qquad (12) \\ & \textstyle\sum_{r,j\,:\,r \neq j} J_\tau\big(\mathbb{A}[r,j]\big) \leqslant \kappa \\ & \Lambda[r,k] + \mathbb{1}(j \neq k) - \Lambda[j,k] \geqslant J_\tau\big(\mathbb{A}[r,j]\big) \quad \text{for all} \quad r, j, k \in \{1, \ldots, p\} \text{ and } r \neq j \end{aligned}$$

Your Main Goal

Despite the fact that the Authors of the original papers derived and implemented in `MLEdag()` the χ^2-asymptotics for the Likelihood Ratio statistics under both testing scenarios, your goal here is to push the **finite-sample** envelope and adapt the two **universal procedures** of Equation 2 to the current framework.
You will be using `MLEdag()` to get the MLEs you need and also to compare the results you get ⇝ install that package!

The Data: Cell Signalling

We will use all the machinery above to dig the multivariate flow cytometry data in Sachs et al. (2005). Suppose we are studying the human immune system, in particular the so called *T helper cells*. To properly understand their normal responses under some specific extracellular stimuli, we can perturb them with a series of chemical stimulatory/inhibitory interventions and then profile the effects of each condition by measuring the expression of relevant proteins and lipids via flow cytometry.

The data collection process is described in Figure 1:

1. each cell is treated as an independent observation;
2. 9 different perturbations were applied to sets of individual cells, namely: `cd3cd28`, `cd3cd28icam2`, `cd3cd28+aktinhib`, `cd3cd28+g0076`, `cd3cd28+psitect`, `cd3cd28+u0126`, `cd3cd28+ly`, `pma`, `b2camp`. The known biological effects of the perturbations employed are described in Table 1 of Sachs et al. (2005);
3. On each cell in each condition, a multiparameter flow cytometer simultaneously records the levels of 11 proteins and lipids, namely: `praf`, `pmek`, `plcg`, `PIP2`, `PIP3`, `p44/42`, `pakts473`, `PKA`, `PKC`, `P38`, and `pjnk`. This simultaneous measurements allow researchers to infer **causal influences** in cellular signalling.

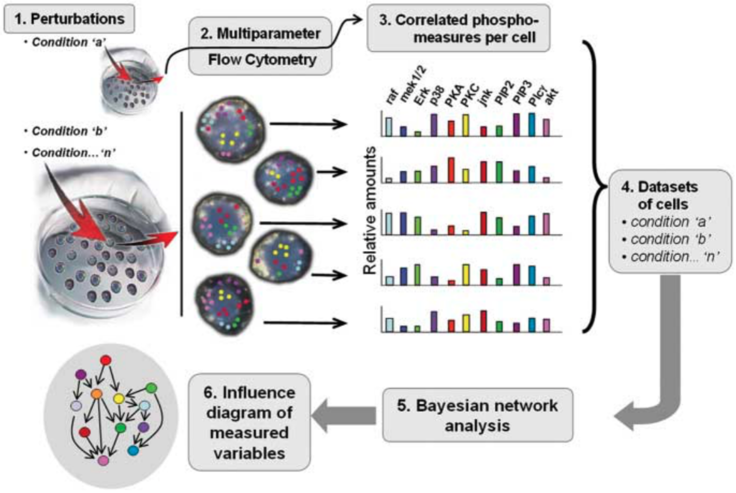


Figure 1: Source: Sachs et al. (2005)

The Exercise: ToDo List

1. Read (really) these notes, and install the `clrdag` package directly from `git`:

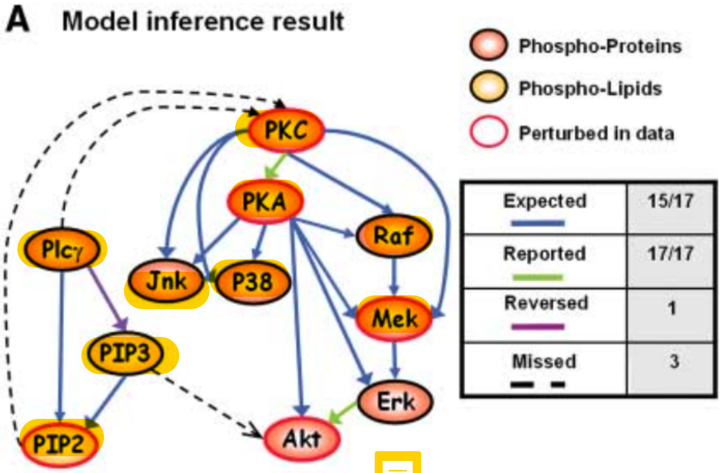
```
install.packages(devtools)
devtools::install_github("chunlinli/clrdag/pkg/clrdag")
```

   Notice that our sparsity parameter κ is `mu` in the main function `MLEdag()`. To install this package on `OSX`, you need to *"brew"* `gcc` and then make it visible to `R`. Please read this and/or contact us.

2. By using the `MLEdag()` function to get constrained and unconstrained MLEs, adapt and implement in `R` at least one of the **universal tests** in Equation 2 to the problem of testing for **graph linkages** and **directed pathway**.

3. Starting from the code in the `examples` folder of the `clrdag` package and trying to find some inspiration from *Example 1* and *Example 2* in Section 5.1 of Li et al., design and run a decent simulation study to check **size** and **power** of your **universal test(s) for linkage**.

Let's now move to cell signalling and flow cytometry. By stacking together the data coming from **all** 9 interventions, Sachs et al. (2005) estimated the DAG in Figure 2 via a data-discretization technique. The edges were categorized as: (i) **expected**, for connections well-established in the literature that have been demonstrated under numerous conditions in multiple model systems; (ii) **reported**, for connections that are not well known, but for which we were able to find at least one literature citation; and (iii) **missing**, which indicates an expected connection that their network failed to find. Of the 17 arcs in their model, 15 were expected, all 17 were either expected or reported, and 3 were missed



Expected	15/17
Reported	17/17
Reversed	1
Missed	3

Figure 2: Source: Sachs et al. (2005)

4. Looking at the estimated DAG in Figure 2, formalize at least 3 linkage-type hypotheses and 1 pathway-type hypothesis that you feel it may be interesting to double-check with your own toolkit. Briefly explain why you find them interesting.

5. Select one specific intervention out of the 9 available and, based on those data only, test your set of hypotheses using both, your **universal procedures** and the asymptotics implemented in the `MLEdag()` function.
   Compare the results also as you let the sparsity parameter κ vary.
   Achtung! Achtung!
   Before running any test, keep in mind we are working under the assumption that the data are *zero-mean Gaussian*! Use basic tools like `boxplot()`, `hist()`, `density()`, `qqplot()`, `pairs()`, to visually check what's going on with each of your 11 variables, and possibly apply suitable transformations like `scale()` and `MASS::boxcox()` to "enforce" it.

6. Repeat the previous analysis augmenting the data by stacking together those coming from different conditions. Despite their heterogeneity, ideally we would like to pull together all the data available as Sachs et al. (2005) did. Nevertheless, if you find handling the "*über-dataset*" computationally infeasible, simply keep stacking data till you can. Compare the results with those in 5. and draw some conclusions. Do you think we need to **adjust for multiplicity** here? Explain.