10 - SHARP and after clustering approaches

TOPICS	
DATE	@May 18, 2022 2:00 PM
LAST EDITED BY	
LAST EDITED TIME	@May 29, 2022 9:38 AM
MADE BY	
PROF	Calogero
Recording	
STATUS	☑

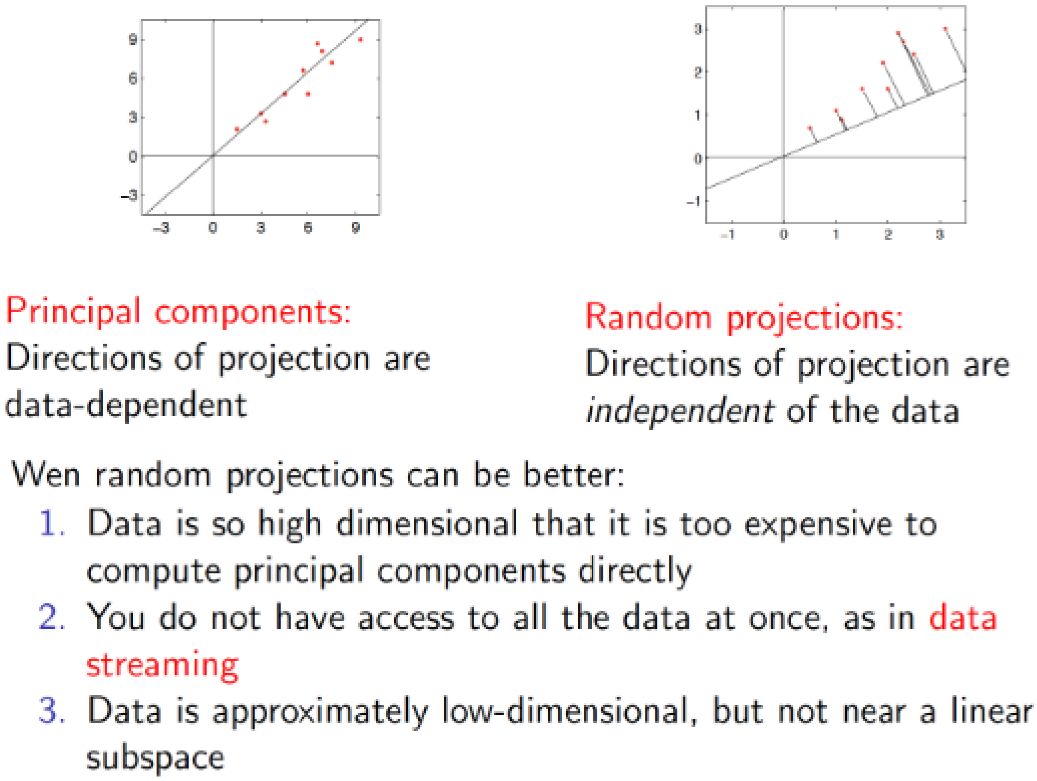
SHARP clustering

▼ **Last year transcipt**

SHARP CLUSTERING
This method follows a completely different approach using randoms projections (a new
method of data reduction), weight-based meta clustering and similarity-based meta clustering.
Random projections
It's a similar process as PCA but while in the PCA approach the projection is done over a line
that is drawn trying to be the closest to the points/data, in random projections this line is
random and therefore the projections are too, it's not necessarily the shortest projection.
It's useful when the data is too big to calculate or estimate the principal components directly.
It also is useful when the data is low-dimensional but not quite a line.
So, the way SHARP works is that fist divides the data set in random blocks, then in each

bloks multiple random projections are done that produce a different data reduction (each
random projection produces a different data reduction). Then these different projections are
clustered together (weight-based meta clustering) to get an aggregation. Then all the
aggregated projections from the different blocks are clustered together with the similarity-based meta-clustering.

SHARP is another type of clustering approach and there are many more that we won’t discuss. SHARP was claimed to be **suitable for large datasets** (up to 1 million cells). It starts by dividing the sample in different subsets and use a data reduction approach that is similar to the principle component analysis, since it does a linear transformation in 2D dimension.



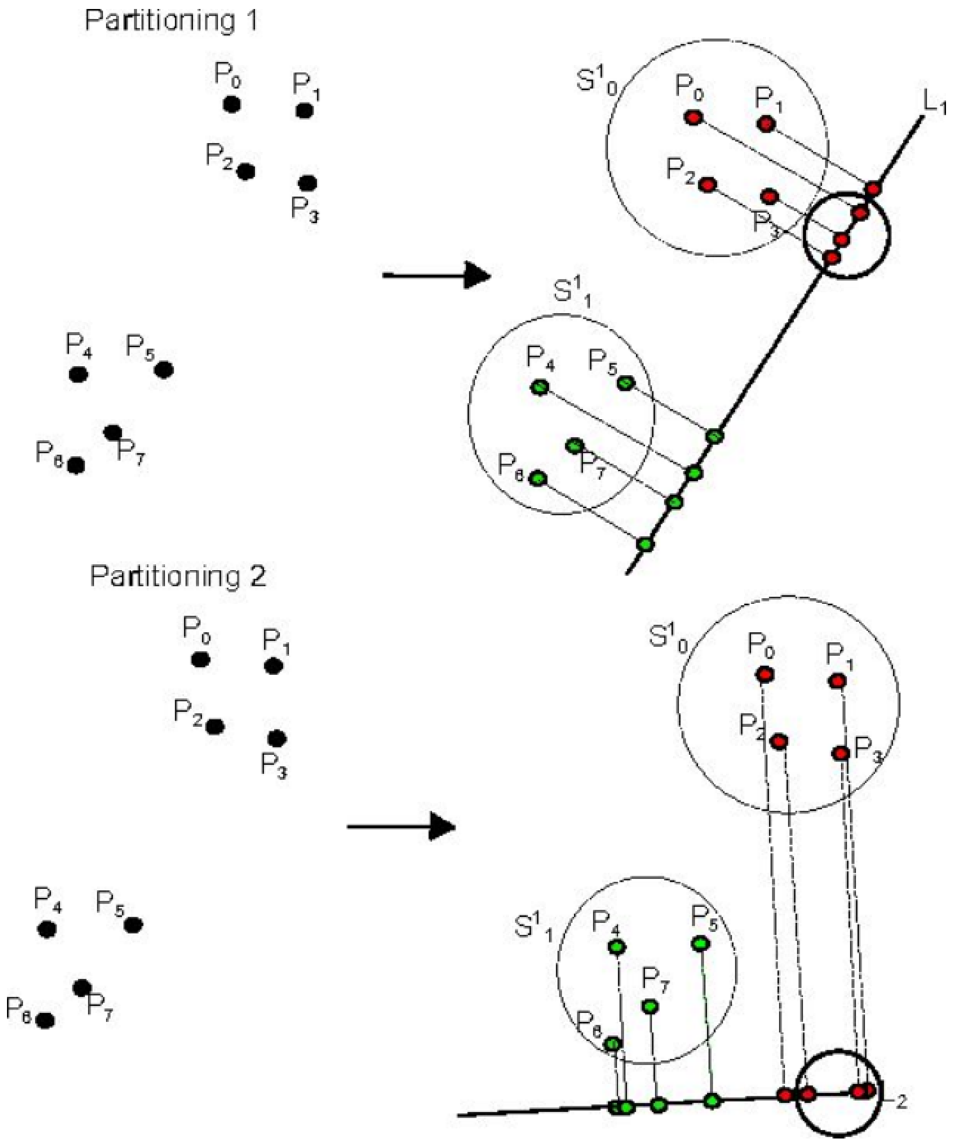
Principal components:
Directions of projection are data-dependent

Random projections:
Directions of projection are *independent* of the data

Wen random projections can be better:

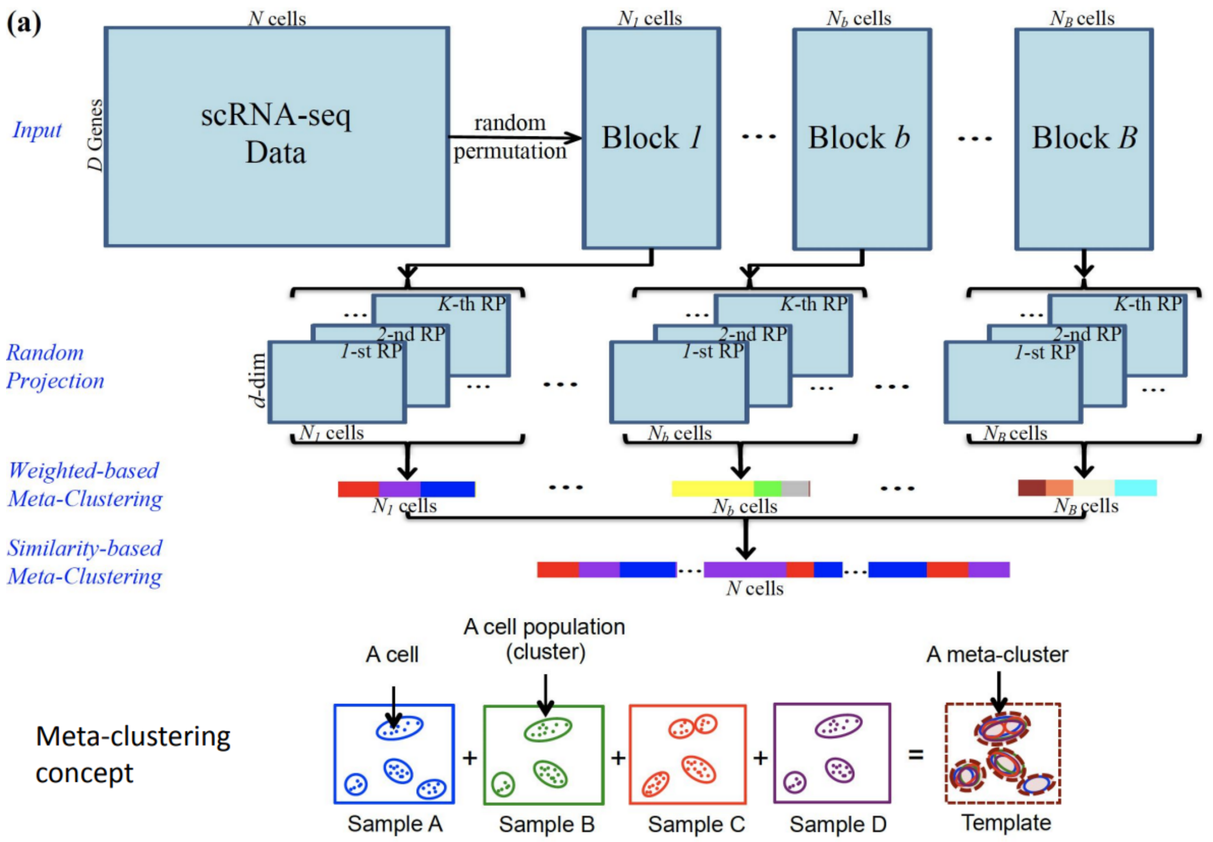
1. Data is so high dimensional that it is too expensive to compute principal components directly
2. You do not have access to all the data at once, as in data streaming
3. Data is approximately low-dimensional, but not near a linear subspace

SHARP hence requires dimensionality reduction which is done on a subset of the data in a similar way to PCA but with some differences: in PCA we are doing a linear transformation of our data to fit it to a lower dimensional space, while in SHARP only a little part of the overall data is used in data reduction. Another important difference is that with PCA we find first the first component representing the majority of the variance of the data and at this point we make the second dimension and all the others orthogonal to this first one. With SHARP we use an arbitrary dimension which is different for each dataset and it will represent only some of the characteristics of the data in this space.



2 subsets of the data are represented over 2 different dimensions

SHARP works using **random projection.** You define a dimension that is totally arbitrary and for each dataset you use different arbitrary dimension. The space is randomly selected. Basically what it does is to partition the data in different dimensions (it's like using a single kernel for each subset of the data).



This **partition of the data before dimensionality reduction allows data compression**: it does a random permutation of the data (= data partitioning in many smaller subsets and these subsets are random, with some cells being present in multiple subsets) and then data reduction is performed on every single subset on multiple different dimensions. The same dataset is represented on multiple ways. Consider one subset: each representation on different random projections will be compared with the others to get a similarity matrix gathering more similar clusters together independently from the projection used. Basically we are putting together all the data that are clustered similarly in different dimensions together in one representation putting together the data that are represented similarly in different dimensionality reduced spaces. This is possible since we have done partially overlapping subsets. This approach of using the same data projected in different dimensionally reduced spaces is called **meta-clustering.**

This is done for every subset, and once we have obtained a unique meta-clustering representation for each subset we do the same for the different subsets obtaining a meta-clustering for the whole dataset.

With meta-clustering we put together the information coming from all the different sub-representations to build a representation of our data that is a sort of an average of all the other sub-representations.

> 📌 There is not an ideal clustering algorithm but everything is a compromise of something and we have to make the choice that better fits our data.

We make multiple projection of our data and in SIMILR we select the better subset in representation of our data. Here we find the elements making the different representations very similar. After this step we do meta clustering to build only one representation with all the information.

After clustering

With bulkRNA seq we can do dimensionality reduction and differential expression (DE) analysis to identify genes that are DE and can be used to mine our data to look for paths and networks characterizing biological events we are trying to study at a molecular point of view.

With SC RNA seq instead, we are making different clusters which are representative of different types of cells inside our sample. Our clusters will represent the clusters of cells with a similar transcriptional profile. In some cases, for example with a mixture of heterogeneous cancer samples, we want to cluster together populations of the same cell type in searching for new populations (new subsets of cells with a peculiar transcriptional profile). Each of these subsets will be characterized by specific markers, which are useful because they represent a way to isolate these cells from the other contexts (cell sorting).

Detecting cluster-specific genes: COMET

▼ **Last year transcript**

COMET
Comet is an application that gives you the possibility to run subset of genes specific for a specific cluster. How the expression of a set of genes is going to be specific for a specific cluster. For example, Gene A and gene B are expressed in different clusters, but the expression of both gene A and B is specific to that one cluster, shown in the picture above.

A very important point in single cell analysis is the possibility of isolating the sub population on cells, expressing and discovering biomarkers that could help me in selecting peculiar subpopulations, and those ones only. There are some cases in which even with the best possible clustering algorithm used, some clusters are completely unable to be generated in a stable form. These clusters cannot be used to do any further analysis, they are not stable.

Comet is a derivative application of the hypergeometric test, normally used for gene ontology enrichment. The interesting approach is that it takes the gene expression information and it transforms it in cluster specific expression. After that it checks if this expression is only represented in that cluster with respect to the others. It looks for enrichments in the top ranked expressed genes in my cluster and looks if those genes behave in a different way in the other clusters. Which are the main players of each cluster, the ones driving the organization on the clustering.
Before talking about all of these, let's have a look at the basic hypergeometric test, also called

Fisher test.
The hypergeometric test is widely used for gene ontology enrichment. Gene ontology, where genes are assigned to a set of predefined bins depending on their functional characteristics, in a hierarchical way. Actually, we'll do gene ontology enrichment later, on comet results.
This is our universe of data, genes are represented by the dots, and the color represents a specific gene ontology term.
We perform a ranking in the way that we're selecting a subpopulation. For example, for single cell, this is clustering. The small square is indicating the list of genes detected as differentially expressed, all the others are not differentially expressed. Usually what is done with gene ontology enrichment is that we take a gene ontology term and we evaluate if there is any enrichment in the list of genes differentially expressed.
Let's consider we're looking at the blue gene ontology term, we see that out of the 12 genes in the list of differentially expressed genes, 8/12 are also blue, with 4/12 not being blue. Outside of the list, only two genes belong to that gene ontology term. By eye it seems that there's an enrichment in the list of differentially expressed genes. By eye does not mean anything, we need statistics to back up our studies. This statistic can be built by using a Contingency Matrix.
It's nothing else than a square matrix in which I put inside the list of the differentially expressed genes and the genes belonging to the gene ontology term I'm working with. The 8 are the genes in the list, expressing the term (being blue), within the list of genes differentially expressed we have four genes that do not belong to the term (blank). Outside we have two genes belonging to the term, but not differentially expressed. All the rest are the ones not belonging to the

term and not differentially expressed genes (26). Once I have the matrix, I can apply an hypergeometric rule (don't learn it by heart). That's nothing else than a fisher test that provides me a pvalue that tells me how significant the enrichment for those gene ontology is in the list of differentially expressed genes.

Its important to define the concept of null hypothesis, this is the set of genes that we call differentially expressed selected randomly from the total population. This is required to guarantee the concept of enrichment.

The hypergeometric distribution is the distribution that is actually describing the probability of k success (random draws for which the object drawn has a specified feature, for example a colored ball) in n draws, without replacement (if I extract one ball from 100, the second extraction will be from 99 balls), from a finite population of size N that contains exactly K objects with that feature, wherein each draw is either a success or failure. If I have all this information, I can try to evaluate, using the fisher exact test if there is any specific enrichment for a specific colored ball (biological function, if we're talking genes).

For example, let's say we have a bag with a finite number of balls inside, and I perform 8 extractions, getting 7 blue balls and 1 red ball. The question is, is having this configuration of color something not expected statistically? Is this some sort of enrichment in the colors of the balls inside the bag? In order to do that I have to associate to this extraction a pvalue and therefore I must know what's in the bag.

I have six different types of colored balls, with the %age of each color different, 5 balls for Y, R, Br; 8 balls of Bl, O, G. Let's suppose we are extracting one red ball and seven blue balls, the order in which I extract the balls doesn't have any meaning (I don't care If I get the red one as the first of fourth ball). In order to calculate what is the probability and the pvalue of getting this exact combination we have to sum of all the combinations possible, as shown in the picture below.

If we would like to calculate the probabilities, that's not difficult. For example, the probability of the first ball being blue is 8/40 (8, all the blue balls; 40, all the balls). The probability of the second one being blue as well will be 7/39 and so on. The probability of getting a red ball after seven blue balls, is 5/33. The overall probability of getting this configuration is the product of all the probabilities of the single events combined. The number is pretty small, as expected, 0.0000000065. But we have to take account of all the other combinations as well that we told before (red one being the second, third, and so on). If we calculate them all, and add them, the number gets to 0.00000053 (the number increased by two orders of magnitude).

We have to get from this number (0.00000053) to a pvalue. The calculation used to calculate the pvalue, was the one above, pretty complicated. Let's try to understand the calculation of the pvalue, using a standard example much easier than a complex like the one we saw just now (colored balls).

The definition of p-value is the probability that a random chance generated the data, or something else that is equal or rarer.

Something equal of having two heads is having two tails, for example. So, the probability of generating HH is 0.25 (1/4), the probability of generating something equal is TT=0.25 (1/4, again), something rarer is not possible, since I only have H or T. So the probability of getting HH = 0.25, the pvalue of getting HH = 0.5 (0.25+0.25; data+equal).

Now instead of performing the experiment once, we did it five times, what's interesting to understand is the pvalue of having 4H 1T. We already saw that the probability of getting 4H1T is the combined probability, 5/32=0.15625. The pvalue is calculated again by having something that is equal like, for example 1H4T, and something rarer, like the probability of getting 5H or 5T.

Now, if we go back to the hypergeometic distribution of our previous experiment (balls) what we have to do is that we start from the initial information, and then adding the probabilities of having similar configurations and more rare ones (8 blue, for example). The whole sum brings my pvalue to 0.01. If I put my threshold of acceptance below 0.05 for example, this combination then will not be there by chance, and there is some sort of enrichment that gives me this kind of characteristic output, based on pvalue.

This is the general idea of the exact Fisher test, how is this the utilized by Comet? Comet performs a slightly different approach to this analysis.

XL-mHG Test (modify version of the hipergeometric test)

XL-mHG test is a powerful test to assess enrichment in ranked lists. I have to provide a ranked list, for example the list of clusters and this tells me if I have some sort of enrichment and which genes are more expressed. The main question in

Is this set of top ranked genes characteristic only of this cluster, and not present in the other ones?

The criticalities are where do I have to put the threshold to define which is the subset of ranked genes are the most important, the first five? The first ten? The first five hundred? I must have some type of way to determine that by myself.

XM-mHG does exactly that. It tries to find a group of elements belonging to an enriched set if they're placed at the top of the ranked list. Normally this raked representation is given by 0 and 1 (1 are interesting elements, 0 are not), binarizing the expression level. One interesting part is that the threshold of

significance is determined by the statistical test itself. So practically it testes all the possible threshold number of genes, giving me the smallest pvalue associated to an enrichment, given that specific threshold.
Maybe with the graph we can understand better. The test starts displaying only 20 elements (these could be genes and the number much larger) and starts by having only the first element (gene) of the ranked list studied, and then takes the first 2, 3, 4, 5, …, down to 20 and for each cutoff it goes and looks for enrichments. To each cutoff, the test assigns a pvalue and only the smallest pvalues will later be taken into consideration.
The strength of this tool is that is finds the threshold for the elements (genes) by itself.
There are some downsides to this test though.
To see these, we have to introduce two scenarios

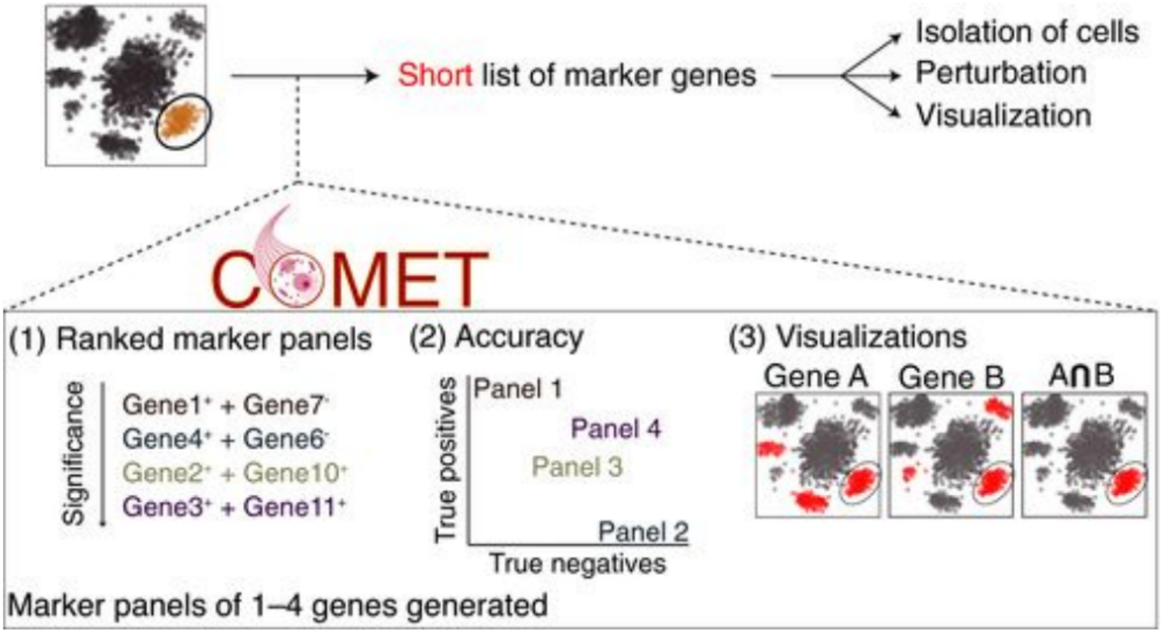
1. The first one has a list of approximately 10000 elements and a moderate enrichment of 1.5 fold (how many fold more did something happen than you would expect by random chance) located in the first half of the group. The significance threshold pvalue is set at 0.01. The first half of the list is analyzed, and performing multiple analysis on this, I'll get for sure some pvalues that are lower to the threshold I put at the beginning. The thing is, that even the
detection of moderate enrichments gives me a "strong" signal, that in a biological environment, could not be significant.

2. Only 1000 elements, with 100 1s distributed randomly into the data. Let us assume that there is no enrichment at all, and all the 1s are as said, randomly distributed into the list, except a few "outliers" at the top (6 out of 100, which is 6% of 1s, very low %age), which are randomly distributed in the first 10 positions. Of course, if I perform my test on the first 20 positions and I find the 1s, even though these are randomly put, the system recognizes them as significant. The mHG is getting sensitive to the outliers 1s. The presence of outliers could give me false positive significant presence of enrichments.
So how are these problems solved in the XL-mHG?
It solves it by the insertion of two parameters: X and L. These compensate the events present in the two scenarios described just above.

3. We can limit the cutoffs tested to the first L ranks. If I take in account all the simulations, I might grab information that are simply due to very faint

enrichments spread allover the upper part of the distribution. L is so the lowest cutoff that is being tested for enrichment (pvalue) .

4. X tells me that only if the upper group of significant elements represent at least the 50% of the population (value selected by you), only in that case I take in account the analysis. For example, if we by principle required at least 15% (X=15) of 1s to be above the cutoff, this would have prevented the six outliers from generating positive results. X is usually put at 15. This doesn’t allow outliers to give false positives. The outliers in the first positions, because of a low percentage of presence, are there just by chance.
   In picture A above, several cells (divided into clusters) are depicted and then analyzed via XL-mHG. For each gene, a threshold for the given clusters is annotated via the XL-mHG test (number of genes in top ranked genes). The test then gives each threshold a pvalue, giving the threshold a measurement of significance. If the threshold is matched to an expression cutoff which is used to binarize (was pvalue of the threshold significant? Ok, then you get a 1).
   Now, lets have an example, given the gene A and B seen before.
   On the right I have the continuous expression measurement, binarized on the basis of the threshold gotten by the XL-mHG test. I do this for both gene A and B, of course the threshold for the two is going to be different. On the middle, the black circle is the intersection between the expression of A and B given the binarizations, you compute what is the overall intersection of each gene with respect to the target cluster. Then you get the plots on true positives versus true negatives (on the right). Then I make gene plots, the expression of A is grabbing the “blue cluster” plus something else, so is the expression of B. The overlapping of both intersecting with the blue cluster, gives the cluster I’m interested in. This means that the co-expression of these two genes is a good expression marker for that specific cluster.
   So, with COMET, with very few genes, we’re able to discriminate between different cell types in real experiments. This helps me optimally identifying sets of genes that are associated to specific subtypes of cells.



COMET is one of the most effective tools to **identify sets of genes that are differentially expressed among the clusters.** COMET is based on an extension of the Hypergeometric test used in gene ontology analyses. This extension is called XL-minimal Hypergeometric test (XL-mHG test).

Gene ontology (GO) is a computer-friendly representation of the genes and their functionality (annotations: molecular function, biological function, subcellular localization). It helps in subsetting our list of DE genes into different processes helping us evaluate better which are the principal biological processes involved in the phenomenon under study.

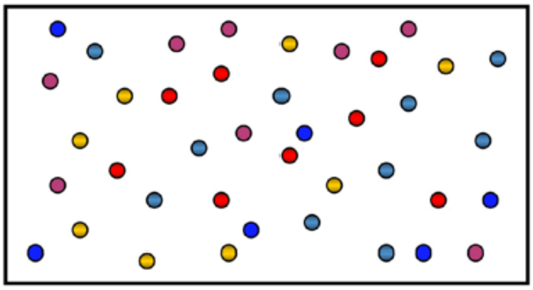
OMICS.net allow us to look at sets of DE genes together with their main TF, miRNA for example. GO allows us to select subsets of characteristics that we think to be related to our biological problem. GO allows to subset our DE genes in a way that they are more appealing to further validation analysis we will do with our data.

There are a lot of gene ontology terms and we need a sort of p-value as score to select the gene ontology terms that really are characteristic of our cluster. This p-value comes from the Hypergeometric test.

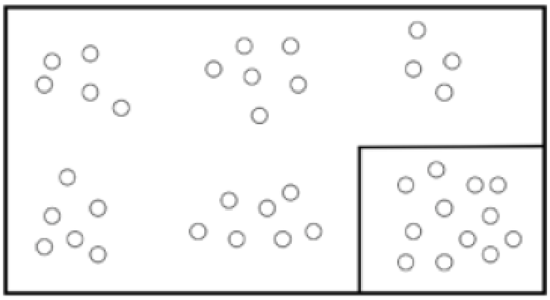
There are many score p-values coming from different kind of statistical tests.

Hypergeometric test for bulkRNAseq: enrichment analysis

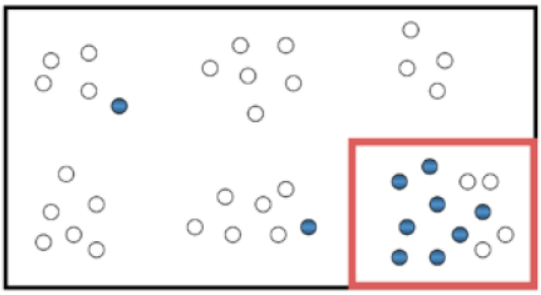
Consider that dots are genes, that are investigated for the DE. The different colors represent different gene ontology terms to which the genes belong.



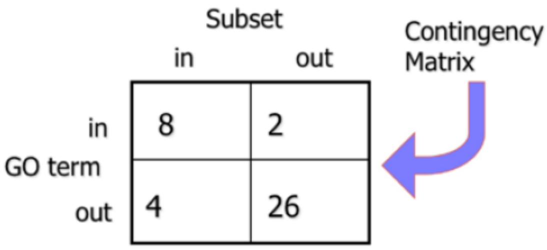
After the detection of DE genes (square on the bottom of the picture), we can overlay to them specific GO terms to evaluate if this term is enriched or not. For example this blue GO term.



By eyes we can see that there are more blue dots here in our subset than everywhere else. We need a p-value to assign statistical significance to this observation.



Applying this **contingency matrix** to our results we can have our p-value. This contingency matrix has 2 options for “being in the subset” and “being under the blue GO term”.



		Subset	
		in	out
GO term	in	8	2
	out	4	26

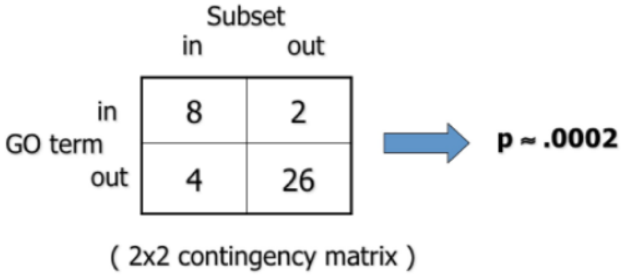
Contingency
Matrix

We apply this matrix to the hypergeometric rule telling the probability of any particular matrix to occur by random selection given that there isn’t any association between the 2 variables. The smaller the smaller the probability that it is due to chance and hence the higher the probability that this term is really enriched within the DE subset of genes.

a	b	a+b
c	d	c+d
a+c	b+d	

$$\frac{\dfrac{(a+c)!}{a!c!}\times\dfrac{(b+d)!}{b!d!}}{\dfrac{n!}{(a+b)!(c+d)!}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}$$

In this case the p-value is 0,002 that is relatively small and so this means that genes belonging to this GO term are enriched in this groups.



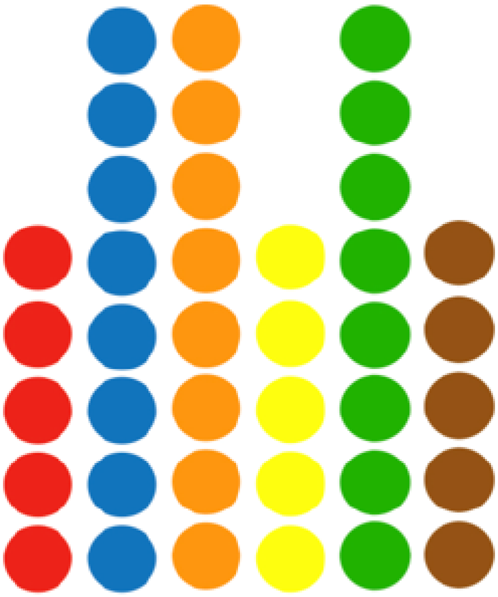
		Subset in	Subset out
GO term	in	8	2
	out	4	26

(2x2 contingency matrix)

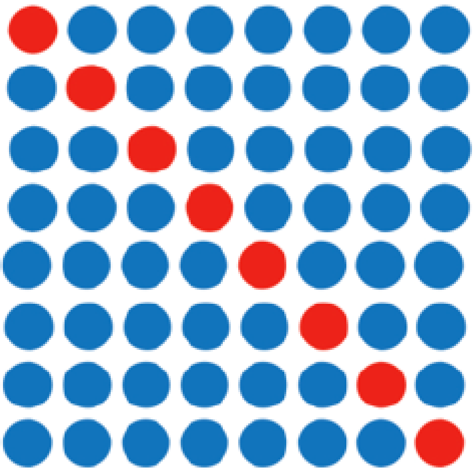
When we are doing an hypergeometric or Fisher test, it is like we have a bag of balls (i.e. the genes in our genome) whose number and color (their associated GO term) is known. You extract for example 7 blue balls and 1 red and you want to know which is the probability of having this kind of output by chance, if you have more blue balls than normal, so you have to compute the p-value.

You need to take care of the respective representation of each color in the full set, like here below.

Red	=	13%
Yellow	=	14%
Orange	=	21%
Green	=	20%
Brown	=	12%
Blue	=	21%



We are calculating the probability to have 7 blue balls and one red. And the probability is little bit different if you consider the red ball in any of the possible positions. The probability of having each of these different extraction pattern is the same though.



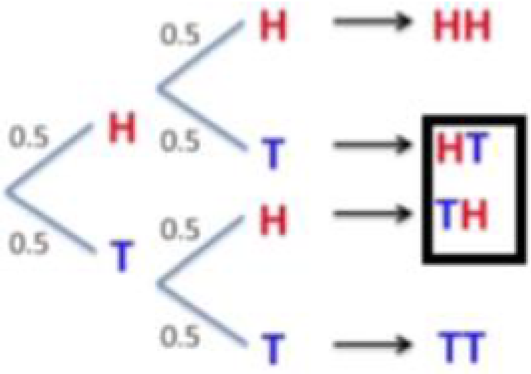
Let’s start by calculating the probability of getting 7 blue balls followed by a single red (last configuration). The probability to have the first blue balls is 8/40, the second blue ball is 7/39 and so on and for the red one the probability is 5/33 because we have already extracted 8 balls from the dataset.

Then we multiply all these probabilities to get the probability of the whole set → 0.000000065

Since we multiply everything, in the end the probability of the extraction is not dependent on the order.

We can repeat the computation of the probability considering any possible order and we obtain 0.00000053

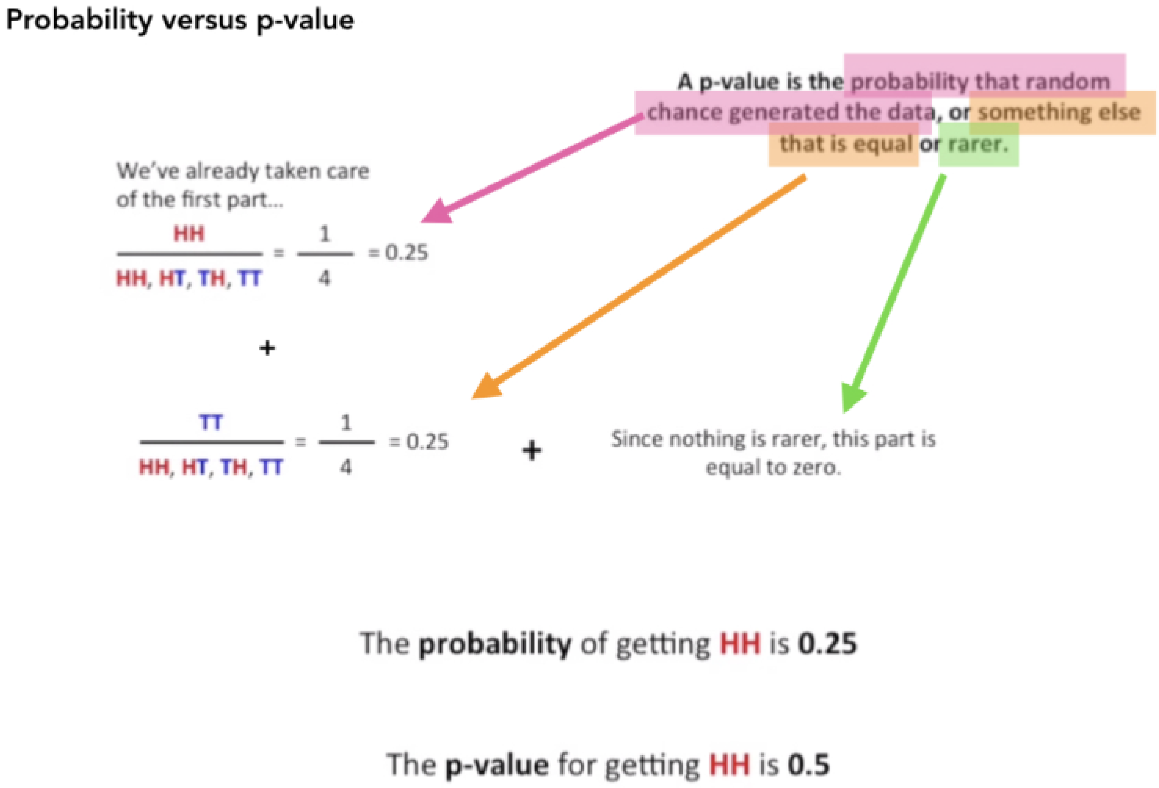
Then we have to move from probabilities to p-values. This is complicated if we look at this kind of data, so we will oversimplify this already simple example to have an idea. Think of a coin: each side of a coin has the same probability to appear and hence the first time the probability to have head or tail is 50%. The probability is the same for both throw. To calculate the occurrence of the heads we have only to consider the total number of outcomes since for each throw the probability of having head is the same.



Now, how to pass to p-values?

📌 The p-value for definition is the sum of the probability that random chance generated the output or something else that is equal (HT or TH in our example) or rarer.

This definition can be applied to retrieve the p-values.



Probability versus p-value

A p-value is the probability that random chance generated the data, or something else that is equal or rarer.

We've already taken care of the first part...

$$\frac{HH}{HH, HT, TH, TT} = \frac{1}{4} = 0.25$$

+

$$\frac{TT}{HH, HT, TH, TT} = \frac{1}{4} = 0.25$$

+

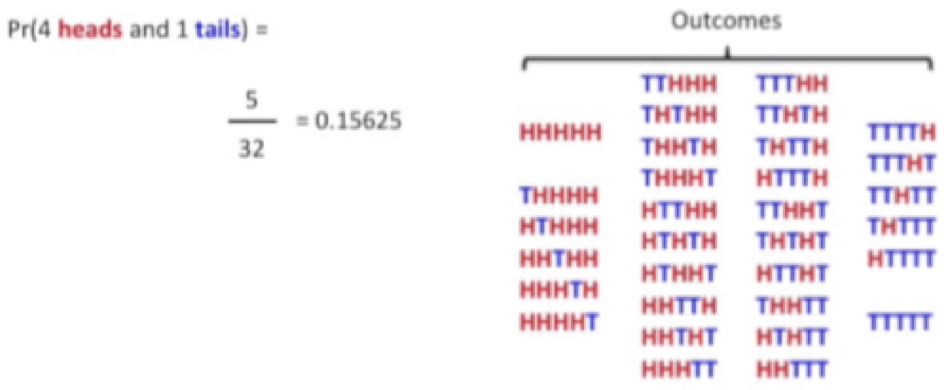
Since nothing is rarer, this part is equal to zero.

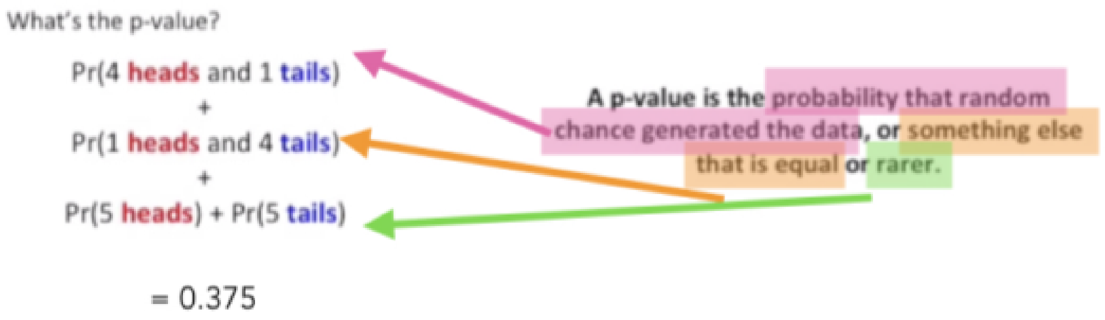
The **probability** of getting HH is **0.25**

The **p-value** for getting HH is **0.5**

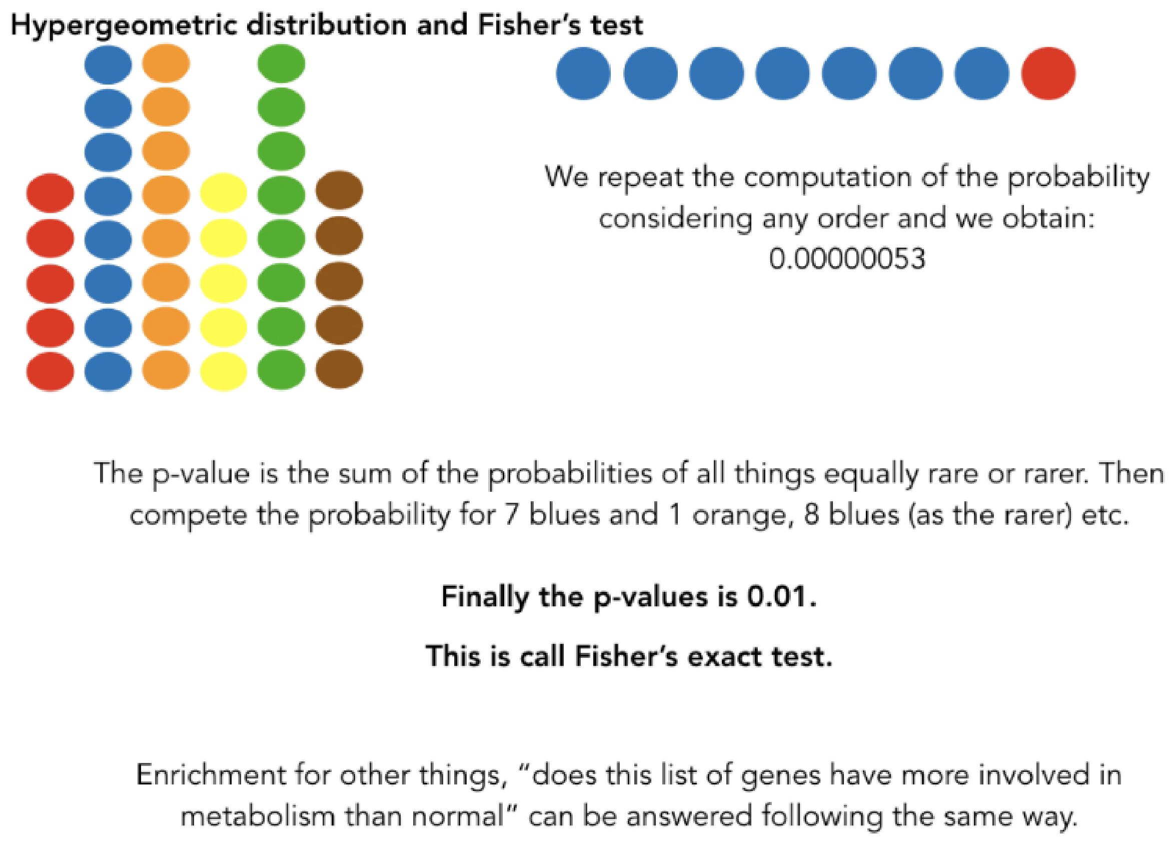
Our probability to have HH is equal to the one of having TT and there is nothing that is rarer: 0,25 + 0,25 = 0.5 which is our p-value.

If we threw the coin more times, when we apply the definition to retrieve the p-values we have even rarer events to consider:




What's the p-value?
Pr(4 heads and 1 tails)
+
Pr(1 heads and 4 tails)
+
Pr(5 heads) + Pr(5 tails)
= 0.375
A p-value is the probability that random chance generated the data, or something else that is equal or rarer.

If we look at our data, and imagine of applying our probability analysis to our datasets of the balls, we will have a p-value of 0.01. This is the output of the **exact Fisher's test** in which we **use the probability of an event to calculate its related p-value.**



Hypergeometric distribution and Fisher's test

We repeat the computation of the probability considering any order and we obtain: 0.00000053

The p-value is the sum of the probabilities of all things equally rare or rarer. Then compete the probability for 7 blues and 1 orange, 8 blues (as the rarer) etc.

Finally the p-values is 0.01.

This is call Fisher's exact test.

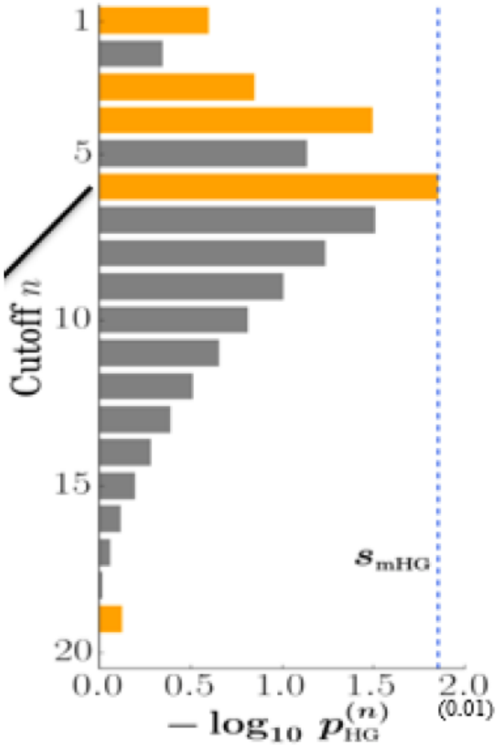
Enrichment for other things, "does this list of genes have more involved in metabolism than normal" can be answered following the same way.

This is how GO enrichment works. We obtain as output a list of GO terms resulting to be enriched in our subset of DE genes. The difference from this and the XL-hypergeometric test is that this latter is used to assess if there is enrichment of a property in ranked lists. We are representing our DE analysis output data as a 0-1 vector (procedure called binarization of the data) depending on if that DE gene has a specific property (i.e. is under a specific GO term or not (1 = yes while 0 = not).

The hypergeometric test sees if the appearance of 1 figures on the top of our vector is significantly increased (i.e. there is an enrichment of that GO term/biological process/whatever) or it is due to chance.

We can also use this test to find specific markers of each cluster: the question in this case is "is this gene significantly more DE in this cluster compared to the others?. If it is we label these cells 1, if not we label them 0. For each gene we obtain a vector made of 0 and 1.

Ex: we study tp53. We have 3 clusters with 3 cells each. We have to binarize expression levels and we obtain that in the first cluster the resulting binary vector is 101, in the second 000 and in the third is 000. We want to see if tp53 is a good marker for the first cluster. The hypergeometric test evaluates if the accumulation of the 2 1s at the top of the vector is significantly different to the other ones or if it is due only by chance.



this test tests all the possible cut-offs to define a p value and selects the one in which it detects the best p-value. In this way the user does not have to specify a fixed cut-off defining "the top of the list" in which the enrichment of 1s has to be tested. This nonparametric approach allows this test to detect enrichment either when there are only a few 1s at the very top of the list as well as when there is only a slight overabundance of 1s within a greater portion of the top list).

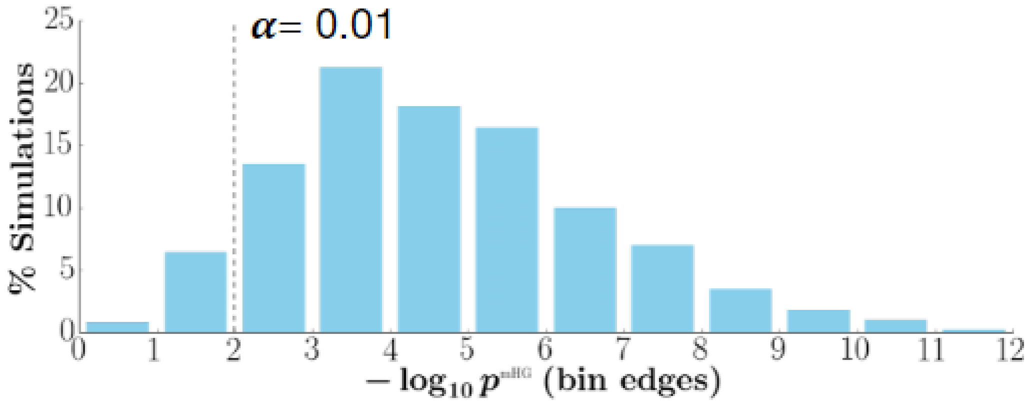
Here the best one is at the threshold = 6 → if we consider a 6-element block above, these 6 elements are so that there is a statistically significant enrichment of at least one of them.

This cannot be done with a normal hypergeometric test but enrichment via COMET requires some adaptations required to compensate some potential errors. Indeed, the kind of hypergeometric used in COMET is not a simple mHG but XL-mHG. The normal mHG test does not exert any control over the cutoffs to be tested for enrichment. The XL-mHG introduces two parameters (X and L) to compensate potential errors depending on the scenario.

XL are two parameters used to compensate potential errors depending on the scenario. They could be errors or biases depending on the domain in which we are running this analysis.

In the first scenario we can imagine having a relatively long list (ex: N=10.000) which has a moderate enrichment (1.5 fold) in the first half. In this case the p-values obtained are not statistically significant because of the large sample: indeed even a relatively small fold enrichment of 1.5 is extremely unlikely to arise by chance given a large enough sample.

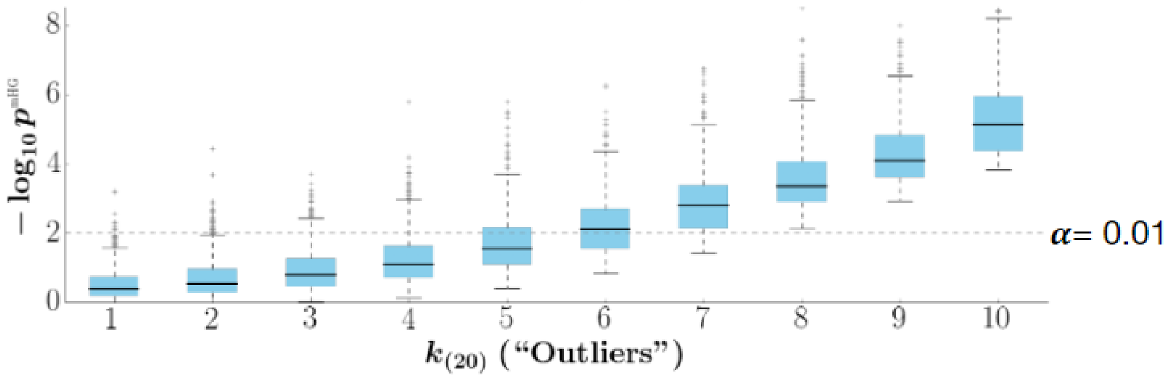
However, in many applications, a slight overrepresentation of 1s among the first half of the list may not represent a very interesting enrichment signal, since weak enrichment among a large part of the list could be artefactual, i.e. arising from a small and potentially unknown bias present in the data.



Moderate enrichment of 1 within the first 5k → in the first 5k terms there is at least one significant enrichment (there are significantly more 1s in the matrix).

This is not always significant from a biological POV. We thus need to make a correction for this scenario.

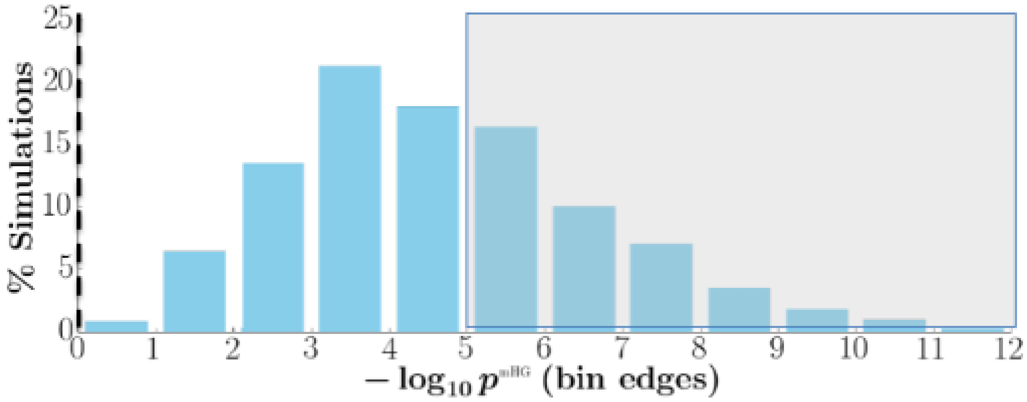
We can have another situation (scenario 2) in which out of 1k elements, we have 100 of these values being 1. The first 20 values are all 1s resulting in a very high overrepresentation of 1s in the very first part of the list. How could this be meaningful biologically? These positive results are based on the high ranking of only 6/100 = 6% of all the 1's in the list. This extreme sensitivity can be thought of as a key feature of the mHG. However, this amazing sensitivity simultaneously makes the mHG vulnerable to outliers.



Hence we need the X and L parameter to restrain the extreme flexibility of the mHG test. They are used to correct what happens in the two scenario presented above.

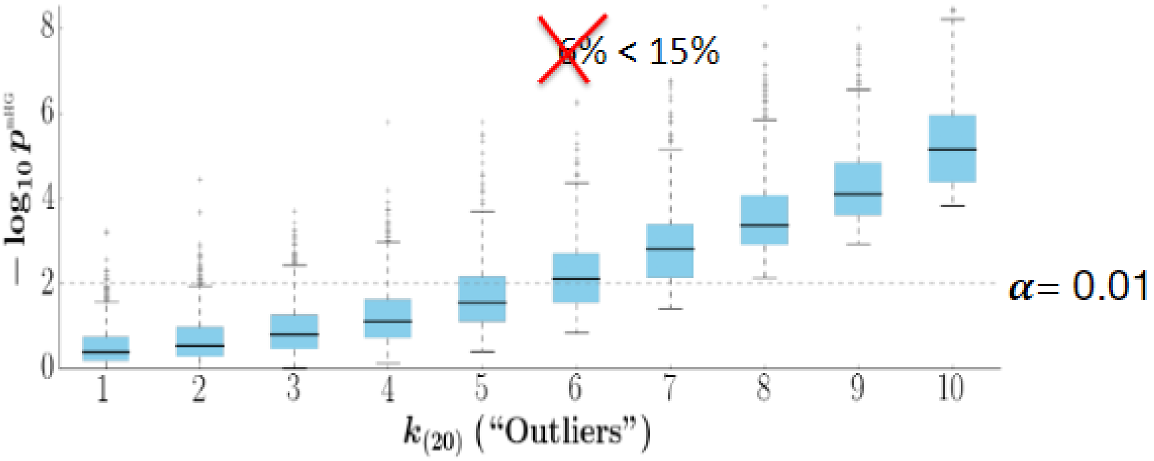
In scenario 1 we were testing even very low cutoffs resulting in a positive test even though the enrichment pattern was artefactual. To avoid this we limit the tested cutoffs to the first L. For example, we might decide that the lowest cut-off at which we would expect to find meaningful enrichment corresponded to N=4 and hence we put L=4. This would significantly reduce the probability of obtaining a significant test result simply because of weak enrichment affecting the top 50% of the list.

L represents the cutoff of the test we are doing (we limit the cutoffs being tested). If within those elements there is a significant p-value this is the result, but if the p-value gets significant only at 1/2 of our distribution this won't be significant. Usually we consider the first quarter of our distribution to calculate the p-value.



In Scenario 2, the high ranking of only 6% of the 1's (the outliers) was sufficient to obtain a positive test result in the majority of cases, even though the remaining 94% of 1's exhibited no enrichment at all.

To improve the robustness of our test, we might decide to ignore all cutoffs that have less than X 1s above them. Had we required at least 15% (X=15) of 1s to be above the cutoff, this would have prevented the six outliers from generating a positive test result.

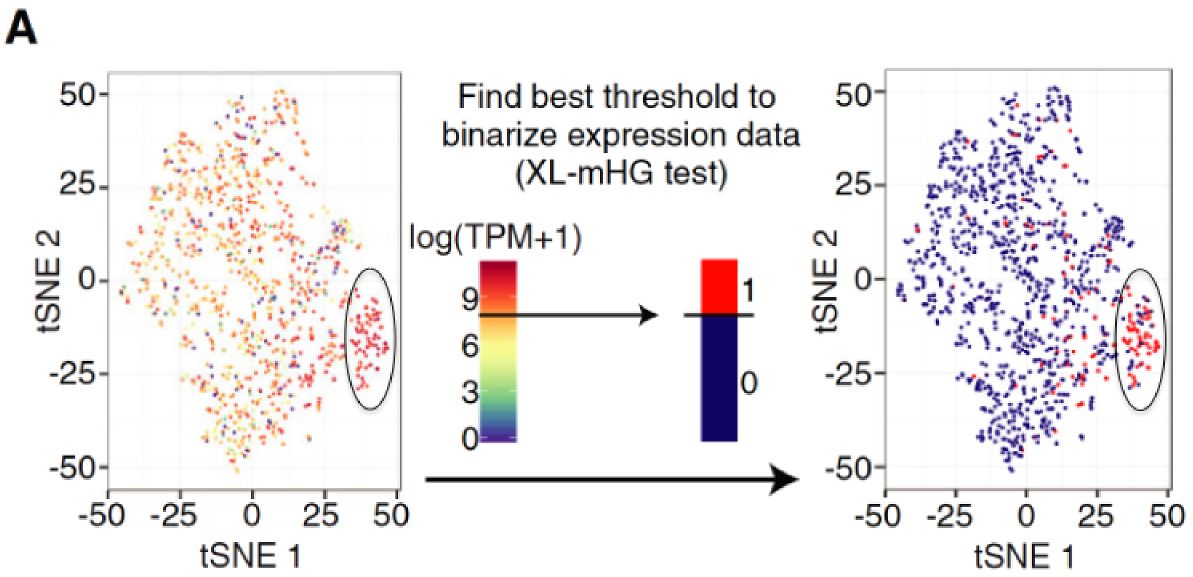


In the end, X and L make the test more robust from the biases.

📌 L is the lowest cutoff (i.e., the largest n) that is being tested for enrichment. X instead refers to the minimum number of 1s that have to be analyzed before anything can be called enrichment.

Be careful: if you narrow too much the window to be tested you are increasing the probability of errors. If you extend a little bit you can see that the p-value is not significant anymore.

In COMET X=15 and L=3, so practically we are looking at a relatively small part of the data to define the representation'.



An illustration of the binarization procedure applied by COMET to each gene in a cluster-specific manner via the non-parametric XL-mHG test. For each gene, an expression threshold of maximal classification strength for the given cluster is annotated with the XL-mHG test. The XL-mHG P-value measures the significance of the chosen threshold index. This threshold index is then matched to an expression cutoff which is used to binarize gene expression values.

Here in this image, the plot on the left represents the log2 TPM of our data. On the basis of gene expression you define a cutoff: here the cutoff is 8 → only the genes with log2 TPM > 8 are set =1 and the other 0. We get many reds inside the cluster and few reds outside. At this point with the XL mHG test we can calculate the p-values associated to this specific partition (the 1s we are looking at are the ones of the cells of the cluster).

You can now define the true positive probability of that marker being associated with that cluster and the true negative probability of that marker not being associated with it. We have now 2 values characterizing the cluster: if all the 1s (represented as red dots in the picture) are present in our cluster and not outside we have that only the cells that are in the cluster are labelled 1 for that gene. In this case there are all true positive and 0 false positives.

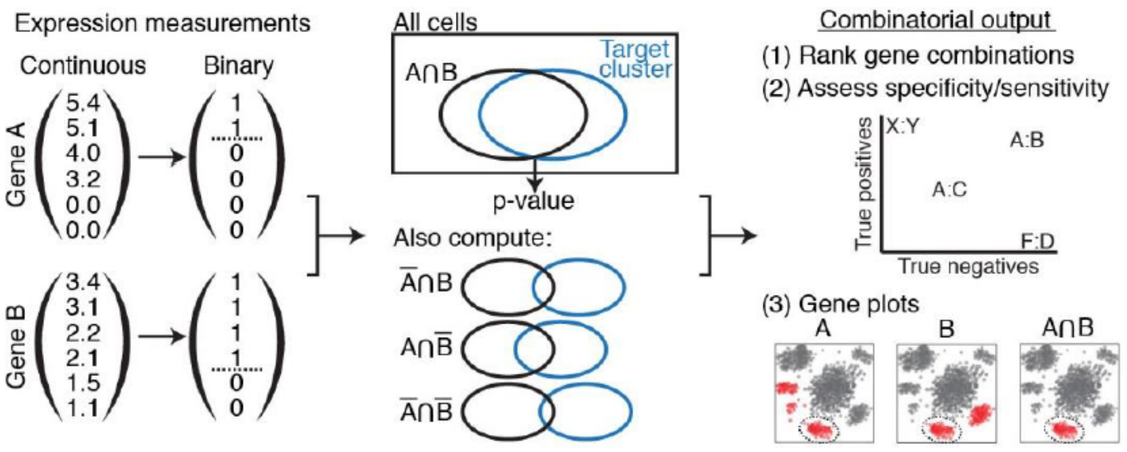
Imagine another case in which we have one cell outside the cluster being red other than the red cells inside. In this case you have 1 false positive only.

> 📌 It is important that during the process of binarization we select a good cutoff trying to have as much 1s inside our cluster and as few 0s inside it, maximising the true positives and true negatives.

This was used to identify genes that are overrepresented in our cluster as markers. Good biomarkers are those which have a good true positive probability and a good true negative probability.

When we have identified 2 or 3 genes for each cluster we are able to identify that kind of cells with that transcriptional profile also with other experimental techniques such as FACS: we can use 2 markers to select these cells.

COMET in the end can be used to identify genes that are markers of specific clusters (i.e. are specifically higher expressed there in my cluster and less in all the others).



Here we have an example in which gene A alone is not very good but combined together with gene B, they can characterize a specific cluster.

The limit for comet are 4 genes.

❗ **When there is very high similarity between clusters, COMET fails**: there are so tiny differences that the system is not able to detect them.

For example, COMET works very well with different cell types where the transcriptional profiles are sufficiently different from one cell type to the other.

In general you use the gene ontology for bulkRNAseq. But you can also do the gene ontology for single cell.

COMET gives a list ranked with probabilities. Taking the top 100 genes is a good representation of our characteristic genes of our cluster on which we can then run GO analysis for enriched biological functions. The top markers are usually related to the cell type which is populating of our cluster.

In single cellwe are defining cell types and look if the % of these cells in the population changes in pathological conditions. We can even find new subtypes, which is not possible with FACS analysis for which you must know which are the markers. You can do FACS later for confirmation.

❗ **About the exam:**

Exam with open questions about theory and maybe some multiple choice. Then some typing errors and a conceptual error to correct. There will be a dataset to analyze by writing a whole script from scratch and maybe some multiple choice.

Could be an exercise in which you have to modify the docker, with the data to be used already inside the docker. This is easier since we can work with bigger datasets without downloading anything.

We can reach up to 33 point → laude