Neuro-genomics - class 6
Statistical significance of functional analysis &
Introduction to sequence variations: detecting DNA mutations and RNA editing events

Statistical significance of functional analysis

Last week we discussed functional analysis, either after clustering or after detecting differentially expressed genes (or both). The question of whether a molecular function, or biological process or cellular component (following the GO categories), is over-represented in a set of genes is a statistical one. Here we will discuss the statistical considerations, following this paper.

The problem description

We consider a total population of genes, e.g. the genes expressed in a RNAseq experiment. The aim is to establish whether the class of the differentially expressed (DE) genes presents an enrichment of the GO category of interest with respect to the total gene population.

Concrete example #1: (note that the numbers in this example are intentionally small)

	Transcription factors (TF)	non-Transcription factors (non-TF)	Total
Class 1 (DE or belong to a specific cluster)	4	3	7 (frequency of TF 0.57)
Class 2 (not DE or outside the specific cluster)	2	11	13 (frequency of TF 0.15)
Total	6	14	20 (frequency of TF 0.3)

In general:

	Category 1 ($\in$GO category)	**Category 2 ($\notin$GO category)**	**Total**
Class 1 (DE)	n_{11}	n_{12}	n_{1+}
Class 2 (not DE)	n_{21}	n_{22}	n_{2+}
Total	n_{+1}	n_{+2}	n

The tables we created are termed contingency tables, and the statistics of the events is well described using the Hypergeometric probability density function:

$$P(n_{11} = x) = \frac{\binom{n_{+1}}{n_{11}} * \binom{n_{+2}}{n_{12}}}{\binom{n}{n_{1+}}}$$

Reminder - the binomial coefficient:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

So this can be written as:

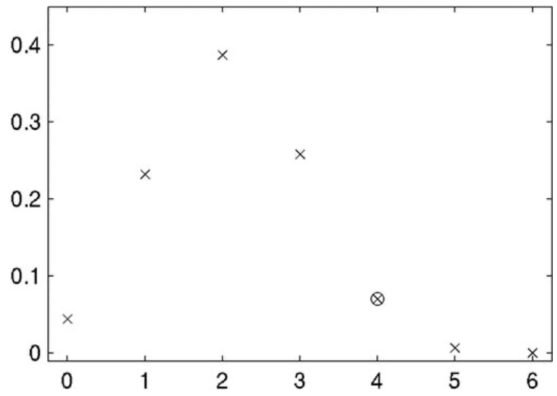
$$P(n_{11}) = \frac{binomial(blue) * binomial(red)}{binomial(green)}$$

For the concrete example #1 above, this gives

$$P(n_{11}) = \frac{\binom{6}{4} * \binom{14}{3}}{\binom{20}{7}} = 0.0704$$

To understand what this probability means, let's first think about the null assumption (H0). The null assumption is that we assume independence between the property of belonging to the GO category of interest and of the property of being differentially expressed (and/or belonging to a cluster).
Under the null assumption, the exact distribution of n_{11} is the hypergeometric distribution, and the probability of getting n_{11}=x is the P(n_{11}=x) given above.

Going back to our concrete example #1, we want to calculate what is the probability of obtaining this many TFs in our DE group given the null assumption. To do that we take a stringent approach - we calculate the probability of obtaining 4 **or more** TFs (i.e. n_{11} >=4) given the null assumption.
P(N_{11}=x) for 0<=x<=6 is shown in the figure below. Note that P(n_{11}=7)=0 and therefore not displayed.



The probability of obtaining 4 or more TFs is:

$$p_{one}(4) = P(N_{11} \geq 4) = P(4) + P(5) + P(6)$$
$$= 7.04 \times 10^{-2} + 7.04 \times 10^{-3} + 1.81 \times 10^{-4} = 7.77 \times 10^{-2}.$$

In this case, using 0.05 level, the null assumption should not be rejected.

Note that for the one-tail Hypergeometric test (as performed here), the calculation is exactly the same as Fisher's exact test.

Performing this calculation using Matlab:

```
>> sum(hygepdf(4:7,20,6,7)) # pdf means probability density function
ans =
   0.0777
% OR:
>> 1-hygecdf(3,20,6,7) # cdf means cumulative distribution function
ans =
   0.0777
```

Performing this calculation using Python: # install scipy first

```
>>> from scipy import stats
>>> from scipy.stats import hypergeom
>>> prb = 1-hypergeom.cdf(3, 20, 6, 7) # cdf means cumulative distribution function
>>> print(prb)
0.07765737874097012
```

Note that if n is large, the hypergeometric distribution can be approximated with the binomial distribution using three parameters: x (below marked as k), n_{1+} (below marked as n), and the rate *p* of TF, which is estimated from the general population: n_{+1}/n.

$$\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

for $k = 0, 1, 2, ..., n$, where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

For the concrete example #1 above, this gives:

$P = \binom{7}{4} * 0.3^4 * 0.7^3 = 0.0972$

Compared to the hypergeometric distribution, computing using binomial distribution is faster and therefore recommended when possible.

Performing this calculation using Matlab (again taking the more stringent approach):

```
>> sum(binopdf(4:7,7,6/20))
ans =
   0.1260
% OR:
>> 1-binocdf(3,7,6/20)
ans =
   0.1260
```

Note that we didn't get the same result as with the hypergeometric distribution (which was 0.0777), because n is not large, and therefore the approximation with the binomial distribution is not valid. However, in many cases the numbers will be larger, and the approximation will hold.

Concrete example #2:

	Transcription factors	non-Transcription factors	Total
Class 1 (DE or belong to a specific cluster)	10	30	40 (frequency of TF 0.25)
Class 2 (not DE or outside the specific cluster)	90	670	760 (frequency of TF 0.12)
Total	100	700	800 (frequency of TF 0.125)

The exact P-value obtained with the hypergeometric distribution is:

```
>> 1-hygecdf(9,800,100,40) % using Matlab
ans =
   0.0198
```

The approximate binomial test leads to a P-value of:

```
>> 1-binocdf(9,40,100/800) % using Matlab
ans =
   0.0227
```

As we expected, the difference between the hypergeometric distribution result and the binomial distribution result are now much smaller.

Note that 1-binocdf(X-1,N,P) is mathematically equivalent to binocdf(N-X,N,1-P), but the latter gives more accurate results due to rounding errors when subtracting something small from 1. So it is recommended to use binocdf(N-X,N,1-P).

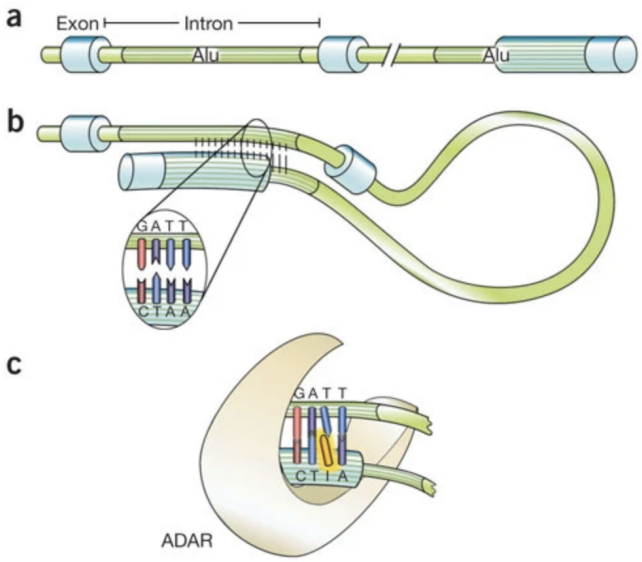
```
>> binocdf(30,40,1-100/800) % using Matlab
ans =
   0.0227
```

Two final notes:

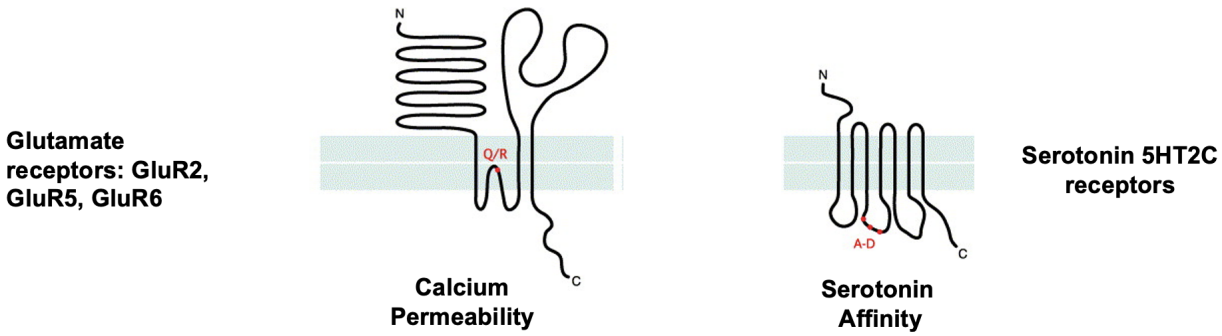
1. It is easy to calculate two-tail p-value, and therefore account for both over-represented and under-represented functions -- the easiest way to do that is by multiplying the P-value described above by a factor of 2.
2. Remember that usually we don't test the statistical significance of just one functional group. Rather, in most cases we test the statistical significance of all possible molecular functions, biological processes and cellular components, or at least the ones which are over-represented and/or under-represented. Therefore, multiple-testing correction should be performed. This can be done, for example, using Bonferroni correction or with Benjamini–Hochberg False discovery rate.

RNA editing in human cells and tissues

The same statistical framework discussed above can be helpful in detecting RNA modifications using RNAseq experiments. Specifically, we can use RNAseq to detect RNA editing events. RNA editing is the process in which the RNA is modified, via enzymes, in specific points compared to the information in the DNA. This can result in different protein sequences, thereby increasing the protein complexity beyond that of the genome. In human cells, the key mechanism for mRNA editing is via the enzyme ADAR, which binds double stranded RNA and changes Adenosine to Inosine. This is known as A-to-I editing. Image from this paper:



In the human brain, there are several examples of how RNA editing can increase the proteins complexity by creating proteins with amino acids that are not encoded in the genome:



Since Inosine is interpreted as Guanosine by the cell machinery and the sequencing machines, A-to-I editing is as if A is modified to G. Therefore, the computational detection of RNA editing is relatively simple - we need to align RNA reads against the genome, and check for locations in which A in the genome is modified into G in the RNA. Remember that this is possible only when full alignment of the reads is performed, using Bowtie2 or HiSAT2 for example, in contrast to pseudo-alignment such as Kallisto.

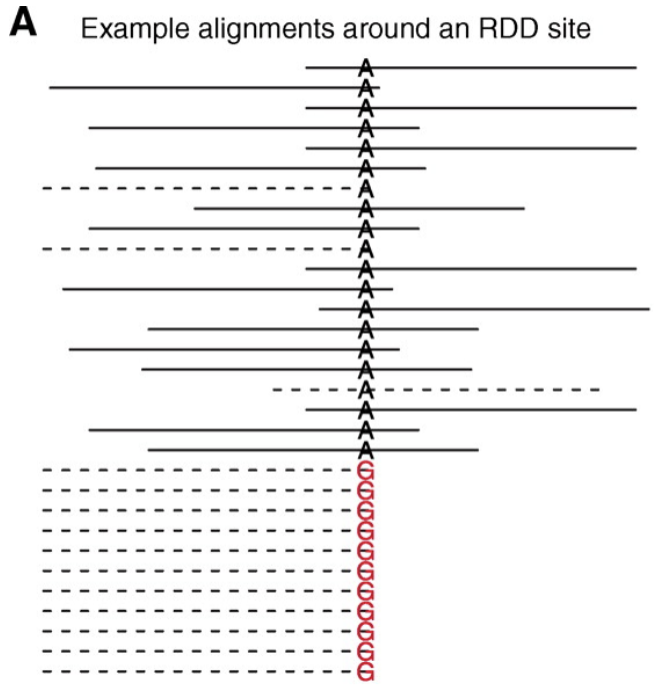
ACTAGTCAACCTCAGGGAATCA
ACTAGTCAACCTCAGGGAATCA
ACTAGTCAACCTCAGGGAATCA
ACTAGTCAACCTCAGGGAATCA
ACTAGTCAACCTCGGGGAATCA
ACTAGTCAACCTCGGGGAATCA
ACTAGTCAACCTCAGGGAATCA
ACTAGTCAACCTCGGGGAATCA
ACTAGTCAACCTCAGGGAATCA

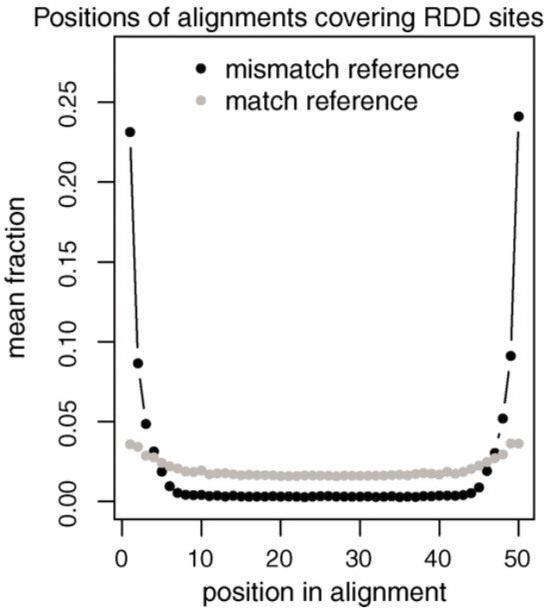
Say that we are examining the aligned RNA reads in a given position, and suppose that we have 50 RNA reads aligned to that position overall. Say that the genome only shows A, and in the RNA we have 10 G and 40 A. Also suppose that the sequencing error rate is 0.1. The sequencing error rate can be estimated empirically by the total number of mismatches divided by the sum of bases sequenced.
The question now is what is the probability that this position shows evidence for RNA editing? The binomial probability of that being a sequencing error is calculated by combining the probability of 10 "successes" (10 G) with the probability of 11, 12, 13 and so on until 50. binocdf(9,50,0.1) is the probability of obtaining 9 or less "successes", so what we need is:
1-binocdf(9,50,0.1)
which is the same as binocdf(40,50,0.9), but the latter is more accurate due to rounding issues.
% Matalb code
>> binocdf(40,50,0.9)
ans =
0.0245
Thus, the P-value is 0.0245, and using 0.05 level we can reject the null assumption (however, remember the multiple-testing problem).

Back in 2011, researchers started to use RNAseq to detect RNA modification in human cells. The first paper, which was published in Science, had the title “Widespread RNA and DNA Sequence Differences in the Human Transcriptome”. As the title suggests, the authors claimed that not only A-to-G modifications are observed in human cells, but rather all 12 possible

modification types (A-to-C, C-to-A etc). It soon became clear that this conclusion resulted from artifacts in the analysis procedure.

As can be seen in the images from this paper (below), most of RNA and DNA Sequence Differences (RDD) were detected at the end of alignments between the RNA reads and the genome. This is a clear indication of alignment errors, which tend to happen at the end of the read:





Statistical significance of DNA modifications

The same statistical framework discussed above can also be helpful in detecting DNA modifications using DNAseq experiments, specifically the detection of somatic point mutation. Although the genome is almost the same between the different cells in our body, somatic mutations will create diversity between the genomes of different cells. Unlike germline mutations, which can be passed on to the descendants, somatic mutations can only affect the organism itself.

Somatic mutations can be caused by endogenous factors, such as errors during DNA replication or reactive oxygen species, or via exogenous factors, such as UV rays, X-rays, and carcinogens such as those found in tobacco smoke. While most of the mutations are corrected by cellular repair mechanisms, some remain. The accumulation of certain mutations over generations of somatic cells can transform a normal cell to cancer cell.

Say that we want to detect DNA modifications in tumor cells, and compare them to normal cells. This is possible by aligning the DNAseq reads against the genome and marking the locations in which there is a mismatch compared to the genome. Again remember that this is possible only when full alignment of the reads is performed, using Bowtie2 or HiSAT2 for example, in contrast to pseudo-alignment such as Kallisto.
Below is a schematic example, from this paper:

Here too some filters are needed to avoid sequencing and alignment errors (again from this paper):