An expedition to The Valley of Genomics

Gianlucca Colangelo[1]

[1]Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires.

Disclaimer: Despite my work is being done in Argentina, this journal will be written in English. The reason is just because of my practicing purpose. That's also a warning from my grammatical or syntax errors.

Entry #1: April 21, Friday: Exploring OMIM.

Objective

Today I would be understanding how the OMIM database is organized. I would see how they classify variables and finally try to visualize them in a compress and all-in-one way.

Objetive

I downloaded two `.txt` files that were brought to me by Nicolas Santiago Nuñez:

- `genes_to_phenotype.txt` tab-delimited
- `phenotype_to_genes.txt` tab-delimited

ID	Sym.	HPO Term	Term ID	Freq. Raw	Freq. HPO	GD Src.	Dis. ID
8192	CLPP	Hypoplasia of the uterus	HP:0000013	-	-	mim2gene	OMIM:614129
8192	CLPP	Primary amenorrhea	HP:0000786	-	-	mim2gene	OMIM:614129

Table 1: Format of genes_to_phenotype.txt file. The first two columns, are ID and Symbols from Gene database. The third and fourth correspond to the Human phenotype ontology. Then we have the frequencies, which is not always given, and finally the gene disease source and the disease ID, which can come from either OMIM or ORPHA

ID	Label	Entrez-id	Entrez-symbol	Add. Info	G-D source	Disease-ID
HP:0000002	Abnormality of body height	55777	MBD5		orphadata	ORPHA:228402
HP:0000002	Abnormality of body height	6598	SMARCB1		orphadata	ORPHA:1465

Table 2: Format of phenotype_to_genes.txt file. The first two stands for ID and Label HPOs. The rest is auto-explicative.

Since I couldn't get yet the raw files from OMIM `mimTitles.txt, genemap2.txt, morbidmap.txt`. I started with the pre-processed listed above, which I think, are from 2020s.

These two files provide the same data. Although the first one has 229862 entries while the second one 944916 entries. The second one is ordered by HPO IDs.

The Human Phenotype Ontology

The HPO is organized as a directed acyclic graph (DAG), where terms (phenotypic abnormalities) are nodes, and edges represent "is-a" relationships between more general and more specific terms. Each term has a unique identifier (HPO ID) and a human-readable

label (HPO label). The ontology is organized into several major subontologies, such as "Phenotypic abnormality," "Mode of inheritance," "Clinical modifier," and "Frequency." We'll be interested on phenotypic abnormality.

Results

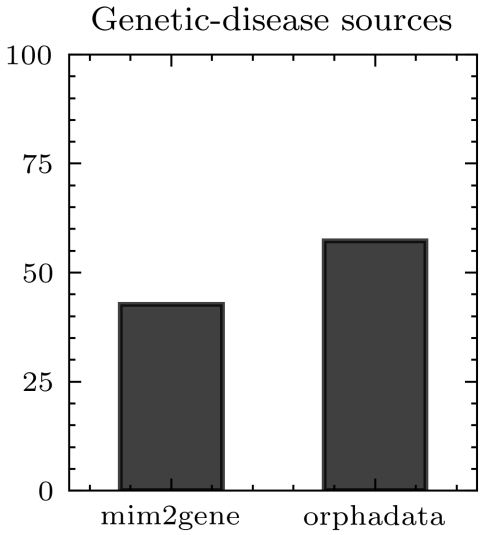


Figure 1: Percentages of genetic-disease sources. mim2gene and orphadata corresponds to OMIM and ORPHA data bases respectively. The total of entries is 944915.

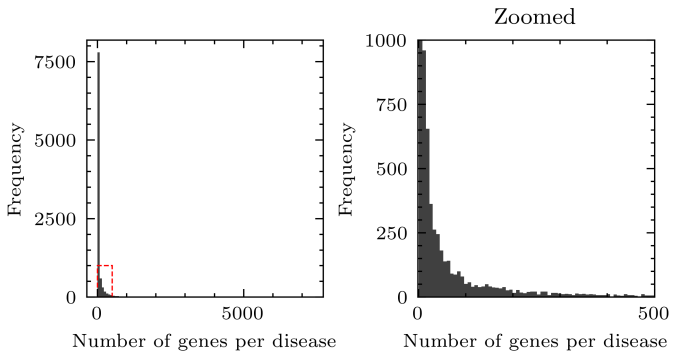


Figure 2: OMIM started as mono-genic mendelian database. But at the moment they also include multiple genes for the same disease. Besides that, up-to 7500 diseases ($\approx$ 79%) in OMIM remain as monogenic. The other 21% are polygenic diseases (caused by more than two genes).

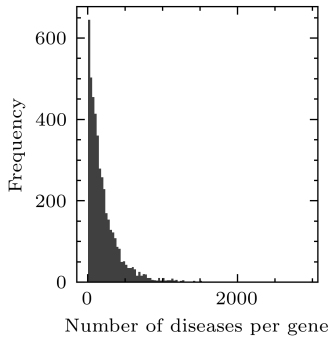


Figure 3: This figure shows the number of diseases caused by a gen. The 65% of diseases are caused by just one gene. And the resting 45% are caused by two o more genes.

Conclusion

The number of diseases is about 9507, while the number of genes are about 4503. There is a ratio of two diseases by gene. Which barely shows a tendency of the complexity of gene

que muestra la complejidad que emerge de los genes, dado que de un mismo gen puede haber distintas enfermedades. Mientras que las enfermedades tienden a tener un solo gen. Esto puede parecer confuso y uno puede pensar que estamos hablando de lo mismo, pero no. Hay más enfermedades que genes, y cuando tomamos un gen, es probable que este esté involucrado en muchas enfermedades. Pero cuando tomamos una enfermedad, esta no está involucrada en muchos genes. Lo cual, pone una facilidad a la hora de predecir de arriba para abajo, es decir, del síntoma al gen, ya que dado un conjunto de síntomas, es probable que se trate del mismo gen. Pero cuando nos paramos en un gen, no sabemos qué síntomas, qué fenotipos, pueden disparar, ya que tienden a ser más.

Entonces, a la hora de diseñar predicciones genotipo-fenotipo, quizás deberíamos preguntarnos si para nuestro poder predictivo no es mejor ir de afuera hacia adentro.

Entry #2: April 22, Saturday: Reconfirming previous results

Objective

I realized i was merging omim and orphadata in the histograms, so maybe the conclusion was biased by the rare diseases. So I will separate OMIM from ORPHA, to see if still has that distribution.

Results

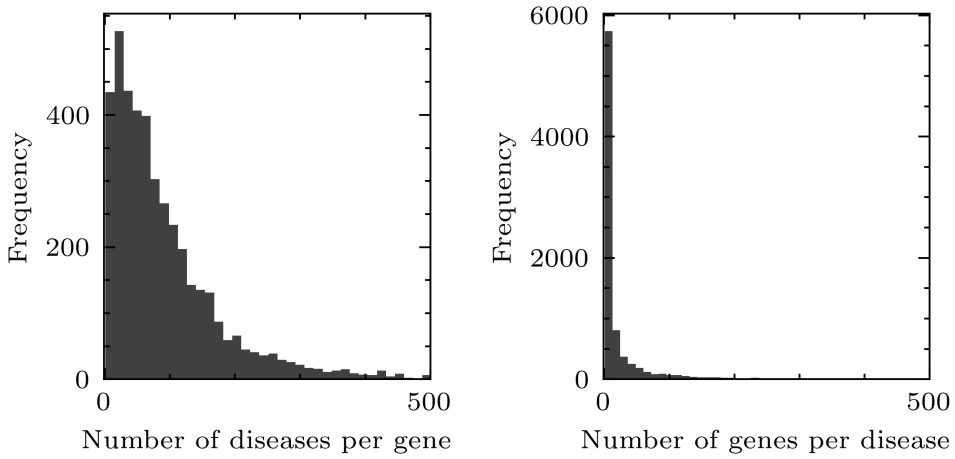


Figure 4: From OMIM database. In this figure we can see clearly how there are many (almost 6000) diseases that are linked to just one gene. While the genes, individually, tend to be linked to more than a few diseases. Generally, there are more diseases than genes.

Conclusions

I can re-confirm the previous hypothesis. That is esier to predict starting from phenotypes than genotypes. It is more probable that a phenotype is linked to just one gene, while is less probable that a gene is linked to just one phenotype or disease.

Entry #3: April 24, Monday: Adding fruits to the salad

Objectives

As the dataframe provided just have practically two variables: gene and phenotype, to exploratory purposes may be good to add some variables like gene length, gene position, gene ontology, etc. It is important to consider that even adding more variables to a model not always result on better predictions, at this point of the work is better to explore more variables to create and evaluate hypothesises.

Methods

I created a function in python that given a list of genes IDs, let me to download them.

```
Entrez.email = "gianlucisnt@gmail.com"


def download_gene_records(gene_ids, file_format="gb", file_name="output",
    max_retries=3, delay=3):
    """
    This function takes a list gene_ids, and iterate downloading them as
    genbank format by default, if you would like fasta, you should put it
    in the second variable file_formate, and in the third variable you
    specify the name of the output file. max and delay are for the http
    errors, very often in entrez api.
    """
    try:
        with open(f"{file_name}.{file_format}", "w") as output_file:
```

```
        i=0
        for gene_id in gene_ids:
            retries = 0
            while retries <= max_retries:
                print(f"Downloading... {i/len(gene_ids)*100}%")
                i+=1
                try:
                    handle = Entrez.efetch(db="gene", id=gene_id, rettype=
                        file_format, retmode="text")
                    record = handle.read()
                    output_file.write(record)
                    handle.close()
                    break
                except HTTPError as e:
                    if e.code == 400:
                        print(f"Error: HTTP Error 400: Bad Request. Retrying
                            ({retries}/{max_retries})...")
                        retries += 1
                        time.sleep(delay)
                        continue
                    else:
                        raise
            else:
                print(f"Error: Failed to download gene ID {gene_id} after {
                    max_retries} retries.", file=sys.stderr)

            # Respect API's usage restrictions by adding a delay between
                requests
            time.sleep(delay)
    except Exception as e:
        print("Error:", e, file=sys.stderr)
        sys.exit(1)
```

Results

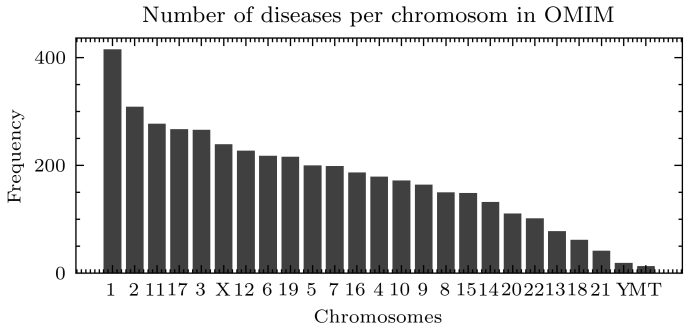


Figure 5: Bar plot of the number of diseases per chromosome from the OMIM database

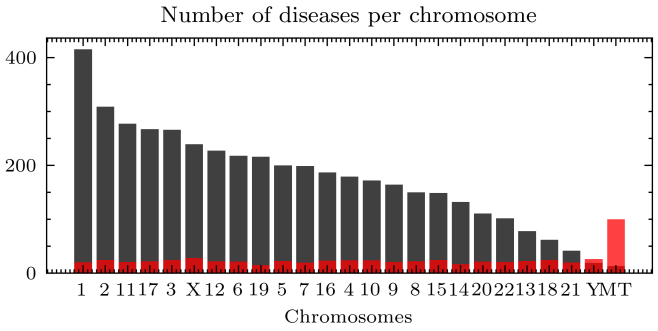


Figure 6: In black, the results of Figure 5, in red, the results normalized.

I basically found that the diseases are not equally distributed by chromosomes, but it's important to consider the limitations of the OMIM database. OMIM focuses primarily on Mendelian disorders and genes, and therefore, the data may not fully represent the complex nature of many diseases.

Beyond that, possible reasons for this result are:

1. Gene density: Some chromosomes may have more genes over total chromosome size.
2. Size of chromosomes: Some chromosomes may be larger.
3. Bias in research: some chromosomes may have been studied more extensively than others.
4. Hotspot mutations

To check this I normalized by number of protein-coding genes based on the last T2T-CHM13v2.0. The results checked the items 1 and 2 above described.

Curiously, the 13 protein-coding genes of mithocondrial dna are reported in OMIM with diseases, that is, the 100%.

These results shouldn't be interpreted as "just 22% of genes in chromosome 1 are diseases related" sino que solo el 22% está reportado tener una variante que cause un fenotipo patogénico, a priori, no sabemos si es porque no están reportados los demás, o porque realmente se introduzca la variante que se introduzca, no van a generar fenotipos patogénicos. En el caso del adn mitocondrial, que son pocos, los 13 genes están reportados. En los demás, habría que ver si hay variantes reportadas en el resto de los genes y ver si realmente al knockear esos genes, no causan fenotipos patogénicos.

En todo caso, para los fines del predictor, sigue siendo información relevante la Figura 5. Ya que cuando tenemos un fenotipo, la cantidad absoluta es la que importa para localizar el gen y no la cantidad relativa. La cantidad relativa nos permitió deshechar la hipótesis de que el cromosoma 1, por ejemplo, tiene una mayor tendencia a generar fenotipos patogénicos.

Además, es importante mencionar que para realizar este gráfico, se hizo un conteo de todos los genes presentes en la base de datos de OMIM con la función `Counter()` de python. De esta manera solo contamos la cantidad de genes diferentes que están asociados a enfermedades.

Entry #4 April 27: HP.obo

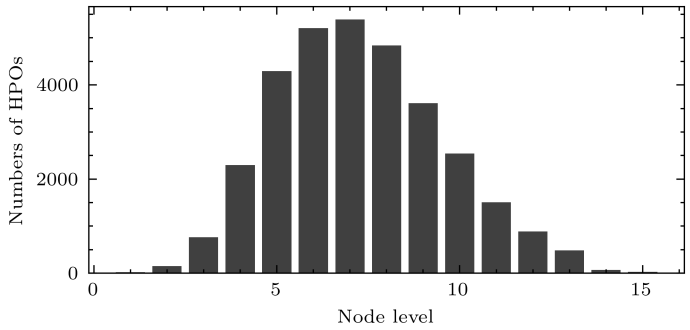


Figure 7: The number of nodes (phenotypes abnotmalities) on each level of the graph. Being the level 0 Phenotypic abnormalities.

Entry # Septiembre 20: Random Forest

Para utilizar un algoritmo de aprendizaje automático, necesitamos más casos por cada enfermedad. El problema es que no tengo los suficientes datos. Es más, ni siquiera tengo datos para todas las enfermedades. Hasta el momento, usando la base de datos de ORPHA, que es la más completa y abarcativa de las enfermedades poco frecuentes, con un modelo simple, que mira la intersección de fenotipos entre los observados y entre las enfermedades registradas en orpha, tenemos un modelo que pone en el TOP 20 a los genes reales. Pero si quisiéramos entrenar un modelo, tenemos el problema de que no poseemos datos reales. Y es un problema porque entre los datos de ORPHA y los casos reales hay un bache bastante grande. Entonces, nuestro objetivo ahora debería ser generar casos sintéticos, repitiendo ciertas propiedades que tienen los casos reales, y validando con el modelo que tenemos, de modo que si los casos sintéticos que creamos, son predichos con la misma curva que los casos reales, los consideramos confiables como para entrenar a random forest con estos casos sintéticos.

Propiedades de los casos reales

- Cantidad neta de casos observados.
- Cantidad relativa (al total de ORPHA para la correspondiente enfermedad) de casos observados.

- Cantidad de casos no específicos (que están a $X > 0$ nodos de distancia de los registrados en ORPHA).
- Distribución de distancias de los nodos.
- Distribución de los fenotipos observados por su peso (ya lo hicimos).
- Cantidad de fenotipos incorrectos

Entry # Octubre 8: Simulación de historias clínicas

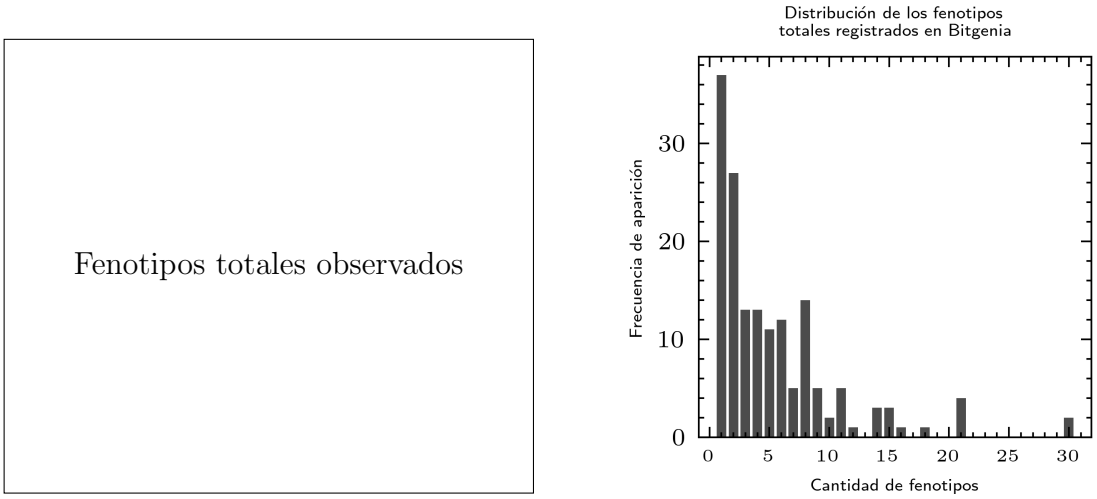


Figure 8: Cantidad total de fenotipos observados.

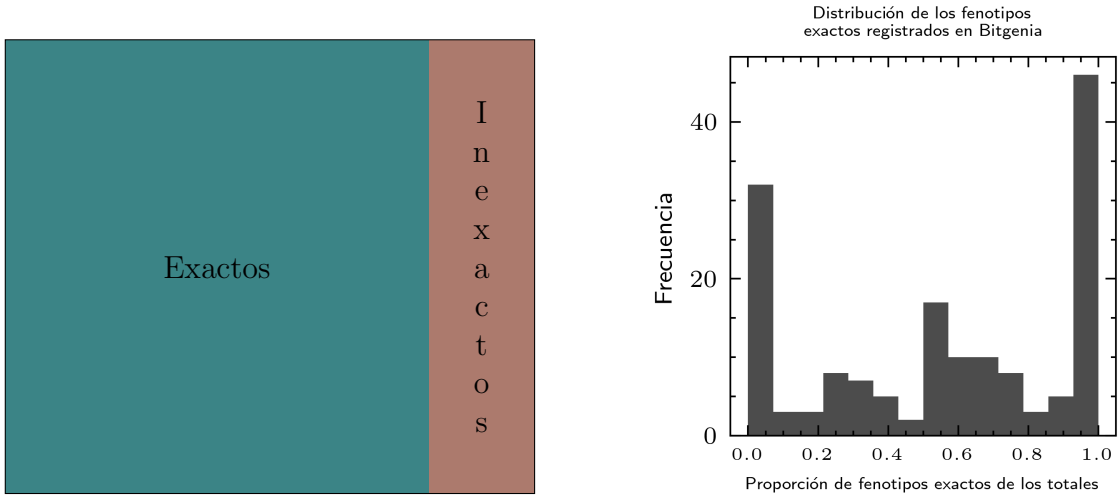


Figure 9: Proporción de exactos en los fenotipos observados totales.

- Obtenemos la cantidad total de observados.
- De esa cantidad total, obtenemos la proporción de exactos.
- La proporción de inexactos es $1 - exactos$.
- De los inexactos, obtenemos la proporción de ambiguos/vagos y de errores.

Que los **datos simulados** tengan el **mismo rendimiento** en el modelo, es una **validación** de que los mismos se están **comportando como los reales**. A su vez, para verificar que no estamos generando exactamente los mismos datos, medimos, la similaridad entre el conjunto de fenotipos reales y simulados para un mismo gen, estas dos validaciones terminan de confirmar la confiabilidad de los datos, ya que:

1. Se comportan como los reales
2. Son diferentes a los reales en los cuales se basaron

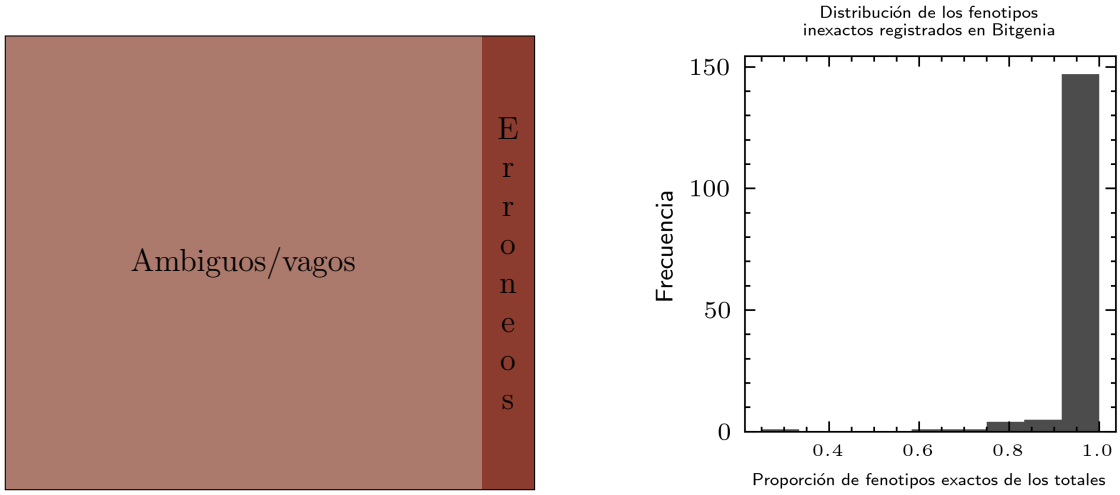
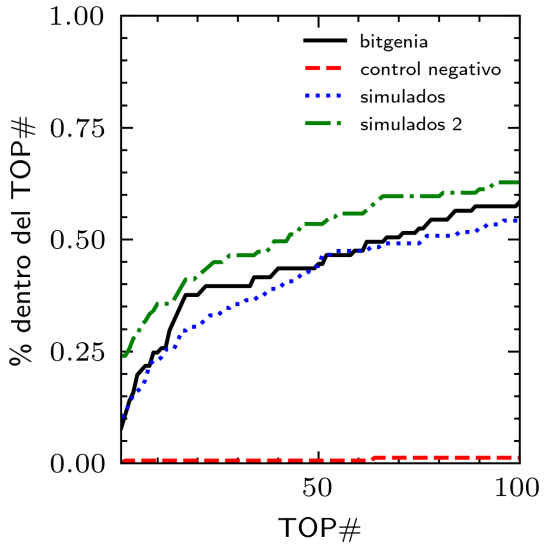


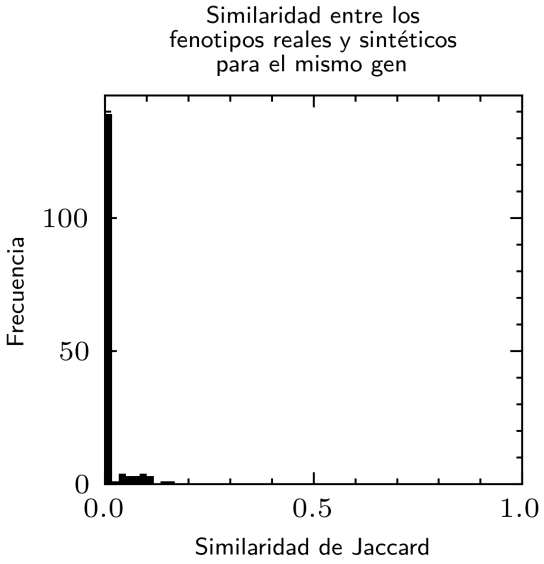
Figure 10: Proporción de ambiguos en los términos inexactos.

Sobre la clasificación de enfermedades

Sea cual sea el modelo que vayamos a usar, tenemos el problema de clasificación. Un modelo multiclasificador de 4000 enfermedades es inviable, por lo que tenemos que clusterizar enfermedades. Para esto, se me ocurren dos maneras de hacerlo: la primera, estadísticamente, que puede ser mediante K-means, a partir de los fenotipos registrados para cada enfermedad, o por proyecciones de redes bipartitas enfermedad-fenotipo. La otra manera es utilizar las clusterizaciones de enfermedades que ya existe en orpha, basada en la anatomía/fisiología de las mismas. En todo caso, quizás haya que realizar todas las clusterizaciones y luego probar cuál funciona mejor para el modelo que vayamos a utilizar.



(a) Simulados 2 elige los exactos a partir de ORPHA mientras que simulados 1 elige los exactos a partir de OMIM. El CN es un conjunto de fenotipos al azar.



(b) Una similaridad de 0 significa que para un mismo gen X, los fenotipos observados son completamente diferentes.

Figure 11