

# Drug Discovery Knowledge Graphs

Combinatorial chemistry has produced a huge number of chemical libraries and data banks which include prospective drugs. Despite all of this progress, the fundamental problem still remains; how do we take advantage of this data to identify the prospective nature of a compound as a vital drug? Traditional methodologies fail to provide a solution to this.

Knowledge graphs, however, provide the framework which can make drug discovery much more efficient, effective and approachable. This radical advancement in technology gives access to model biological knowledge complexity as it is found at its core. With concepts such as hyper relationships, type hierarchies, automated reasoning and analytics we can finally model, represent, and query biological knowledge at an unprecedented scale.

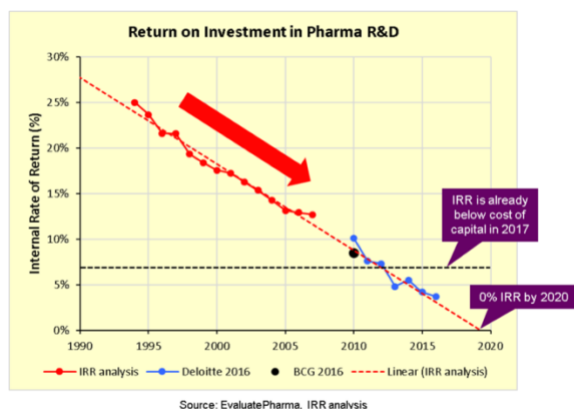
But before we delve into the methodology to create a drug discovery knowledge graph, let us look at what methodologies exist today and why such new techniques are required.

## Drug Discovery and its Shortcomings

Drug Discovery, as the name suggests, is the process by which new medications are discovered. It involves a wide range of scientific disciplines, including biology, chemistry and pharmacology.

Historically, drugs have been discovered through identifying the active ingredient from traditional remedies or just serendipitously. Later, chemical libraries of synthetic small molecules, natural products or extracts were screened in intact cells or whole organisms to identify substances that have a desirable therapeutic effect in a process known as classical/forward pharmacology or phenotypic drug discovery.

Since sequencing of the human genome, which allowed the rapid cloning and synthesis of large quantities of purified proteins; it has become common practice to use high throughput screening of large compound libraries against isolated biological targets which are hypothesised to be disease modifying in a process known as reverse pharmacology or target-based drug discovery. Hits from these screens are then tested in cells and then in animals for efficacy.



The above-mentioned processes may seem systematically effective, however are not as efficient and economical as desired. For a drug to be discovered, on average, it takes approximately 14 years and around 2 Billion US dollars. To add to those numbers, there is a greater than 90 percentage failure rate associated in drug discovery.

With billions of dollars being spent on finding new drugs, with very low outcomes, it has, therefore, become essential to develop newer and more

intelligent ways to do so. To illustrate this need, the diagram above depicts the trend of the return on pharmaceutical investments.

## What are the Challenges in Data Driven Drug Discovery?

While studying the various approaches that leverage data in drug discovery, I found several challenges of achieving the insights that would accelerate and assist the industry. These can be summarised by the following three points:

### 1. Integration — Difficult to ingest and integrate complex networks of biological data

The first challenge starts with the raw data. This data, which pertains to biological concepts and relationships, is scattered all over the world and is produced at an unprecedented rate by a multitude of institutions in various formats and standards. This is also known as the [big data disruption paradigm](#), where high throughput data pipelines are creating bottlenecks to the analysis and processing of that data. It is extremely difficult to ingest and integrate these multi-format and disparate data sets.

### 2. Normalisation — Difficult to contextualise relations within biomedical data

The second challenge stems from the fact that the raw data contained in these data sets have no structure. This lack of structure makes it difficult to maintain and assure integrity, accuracy and consistency over this data. It also causes a lack of control over the validity of data when integrating such heterogeneous data sources. Taken together, this makes it hard to contextualise or understand the relationships contained within the data.

### 3. Discovery — Difficult to investigate insights over a magnitude of data in a scalable way

Finally, due to the magnitude of data, it becomes extremely tedious to generate or investigate insights in a scalable way. Of course, valuable insights can be discovered manually for single data instances, but such an approach is impossible to scale across millions of data points. Moreover, in most cases, doing this manually is simply practically impossible. What do we do then?

## How do we Address These Challenges?

With these problems in mind, we can think of potential solutions that address those challenges. Based on my research, the below is what I suggest:

Integration	Normalisation	Discovery
Ingest and integrate complex networks of biological data into one collection— a knowledge graph	Impose an explicit structure to contextualise the relations between compound, gene and disease networks	Use automated reasoning and analytics to discover and interpret potential targets and drug candidates

Having identified a template to the solutions of the previously listed challenges, I wondered whether there was any one technology out there, that encompassed all three points?

Well, to my luck, [Grakn](#) solves all of these.

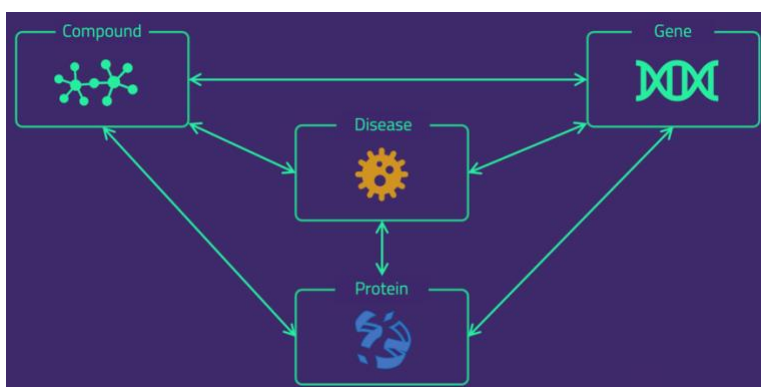
If you're unfamiliar with this technology, Grakn is an intelligent database in the form of a knowledge graph to organise complex networks of data. It contains a knowledge representation system based on [hyper-graphs](#); enabling the modelling of every complex biological relationship. This knowledge representation system is then interpreted by an automated reasoning engine, which performs reasoning in real-time. This software gets exposed to the user in the form of a flexible and easily understood query language — [Graql](#).

## Building a *Drug Discovery Knowledge Graph*?

But how do we actually go about building a drug discovery knowledge graph using Grakn?

### Step 1: Identify the Right Data

Studying the various types and instances of data, I found it beneficial to leverage and navigate the complex relationships between compounds, genes, diseases and proteins.



To augment the above data types, we can also incorporate other types of data which may enrich our knowledge graph as well as provide more powerful insights. This data may entail various biological pathway data or even [text mined medical literature](#) (which can be used to connect certain research studies to give confidence to our insights).

Once we have the raw data we want for our application, we need to find reliable sources where we can retrieve this data from. The following list exhibits some of the sources that may be used to download the raw data pertaining to drug discovery:

1. Gene Ontology
2. NCBI
3. ClinVar
4. CTD
5. DisGeNET
6. Gene Expression Omnibus
7. IntAct
8. PubMed

## Step 2: Normalising Raw Data

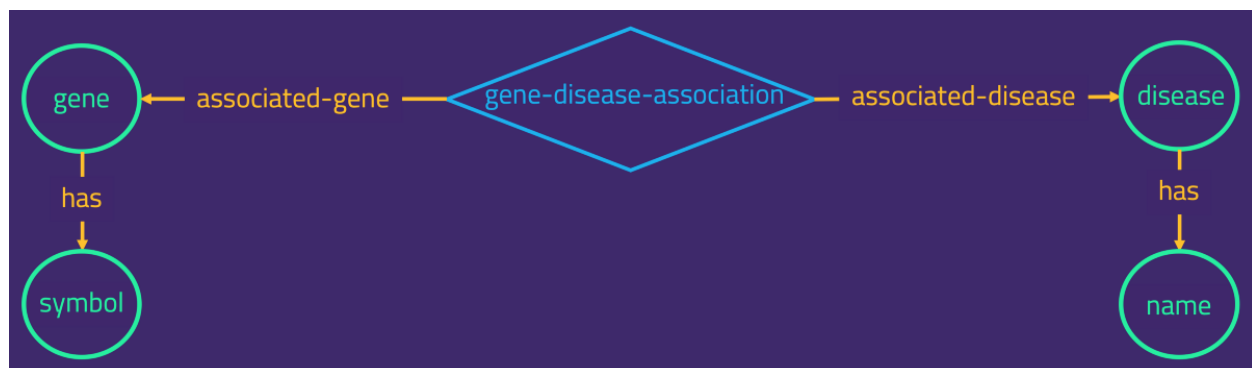
Now that we have raw data, we need to tackle the second problem; that of normalisation. To this end, Grakn utilises the [entity-relationship model](#) to group each concept into either an entity, attribute, or relationship. This means that all we have to do is to map each concept to a [schema concept type](#), and recognise the relationships between them. Let us look at an example to demonstrate how we would go about doing this.

First, let us assume we have the following two raw data sets representing instances of genes in one data set, and another about diseases:

Gene			
gene-id	symbol	...	...
...	...	...	...

Diseases			
disease-id	name	...	...
...	...	...	...

If we were to visualise the above two concepts and the relations between them, it would look like the following:



This structure can be represented using Graql as follows:

```
gene sub entity,  
  has symbol  
  plays associated-gene;  
  
disease sub entity,  
  has name,  
  plays associated-disease;  
  
gene-disease-association sub relationship,  
  relates associated-gene,  
  relates associated-disease;  
  
symbol sub attribute, datatype string;  
name sub attribute, datatype string;
```


We recognised a gene and a disease as an entity, having symbol and name respectively. Then, we defined a relationship between the gene and person that we called gene-disease-association: where the role-players in the relationships are the associated-gene and diagnosed-disease.

In order to constrain the attributes of each concept we also need to denote what data type they adhere to. In this case, we defined them string type.


### Step 3: Migrate into Grakn

Now that we have the data from our sources, as well as a structure imposed on this data, the next step is to migrate this into Grakn. Please note that there are many different ways to do migration, but here I would like to specifically touch on how we would go about using [Java](#), [NodeJS](#) and [python](#).


For this, we can easily use any of these languages to read/parse the raw data file and iterate over each entry within those files. The image below depicts how to insert a single instance of a gene with a name and symbol into Grakn, using any of these three languages:



```
InsertQuery query = Graql.insert(  
  var("$g").isa("gene").has("gene-id", "1234").has("symbol", "ABC")  
);
```



```
const query = await transaction.query (  
  "insert $g isa gene, has gene-id " 1234", has symbol " ABC"; "  
);
```



```
query = transaction.query(  
  "insert $g isa gene, has gene-id " 1234", has symbol " ABC"; "  
)
```

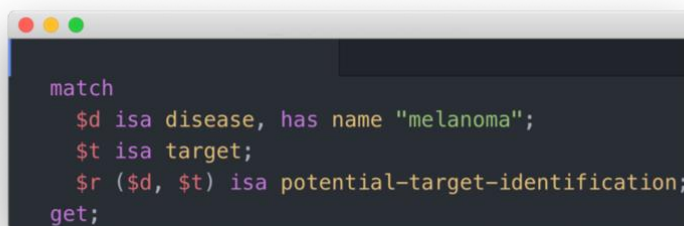
To learn more about [migrating data](#) into Grakn, make sure to read this article: [Modelling & Migrating Big Biological Data with Grakn](#).

#### Step 4: Discover Insights That Scale

After migration, we can start to discover new insights. Discovering insights refers to finding new data that may be valuable to what we are trying to accomplish. In order to do that, we need to first look or ask for something. In other words, we start with a question — the questions we ask to find answers to in drug discovery.

Let us look at an example, and see how our drug discovery knowledge graph may provide answers to them:

*Question: What are the potential targets of the disease Melanoma?*



```
match
  $d isa disease, has name "melanoma";
  $t isa target;
  $r ($d, $t) isa potential-target-identification;
get;
```

Answer:

The answer returns is a list of proteins which are usually potential targets. For proteins to be identified as viable targets they must be related to the disease in some way. Let us say that in this case, they were part of a protein-protein-interaction pathway:

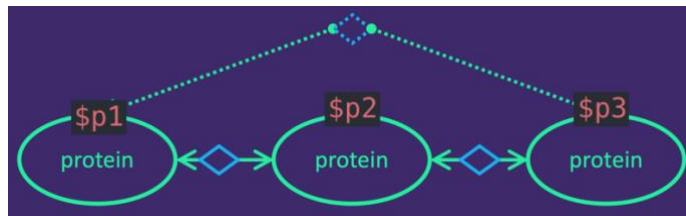


The question that then arises is — how did Grakn recognise the proteins that were not directly related to the disease?

Well, Grakn utilises the power of automated reasoning, in the form of rules, to infer that all proteins that fall in a pathway leading to a disease can be identified as drug targets. Two rules were encoded in the knowledge graph that enabled this.

```
protein-protein-transitivity sub rule,
when {
  $p1 isa protein; $p2 isa protein; $p3 isa protein;
  ($p1, $p2) isa protein-protein-association;
  ($p2, $p3) isa protein-protein-association;
  $p1 != $p3
}, then {
  ($p1, $p3) isa protein-protein-association;
};
```

First, we used a rule that formed the protein-protein-association relations between proteins that had a transitive relation between them. The Graql syntax on the left shows this rule, while the following diagram displays how this rule works:

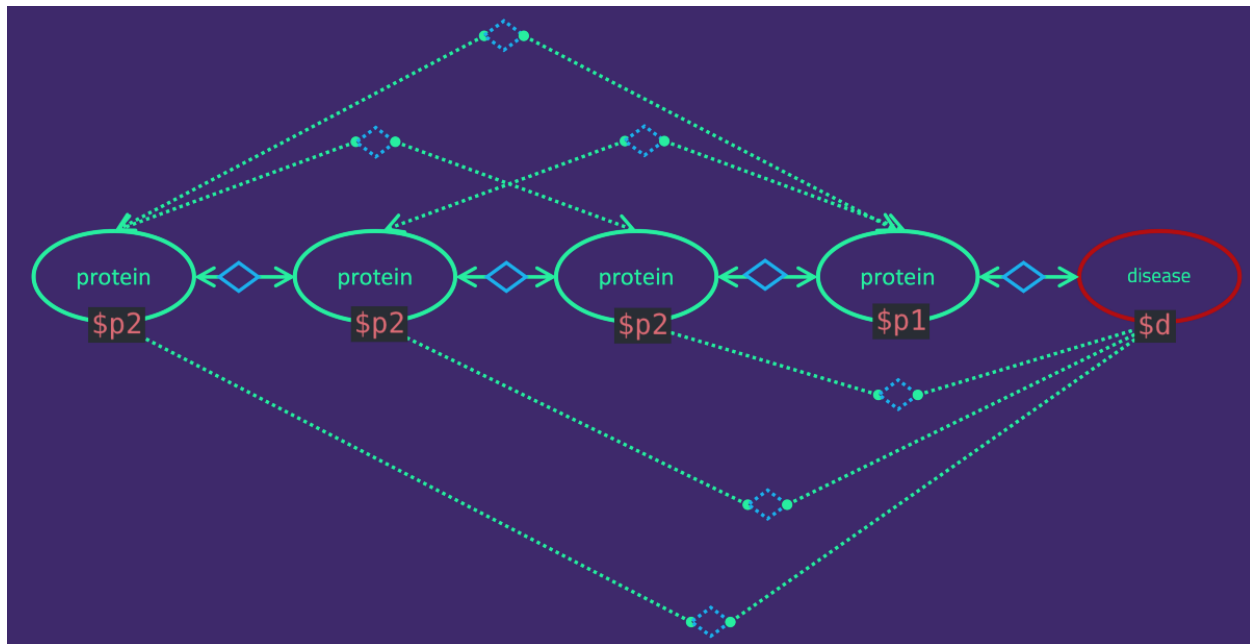


The dotted relation on the top is the inferred protein-protein-association relation that was created by Grakn. This rule resulted from all the proteins that are in a pathway and are linked to each other.

The second rule used a somewhat similar flavour of rules to form the initial protein-target-identification relation between the disease under observation and the potential target proteins.

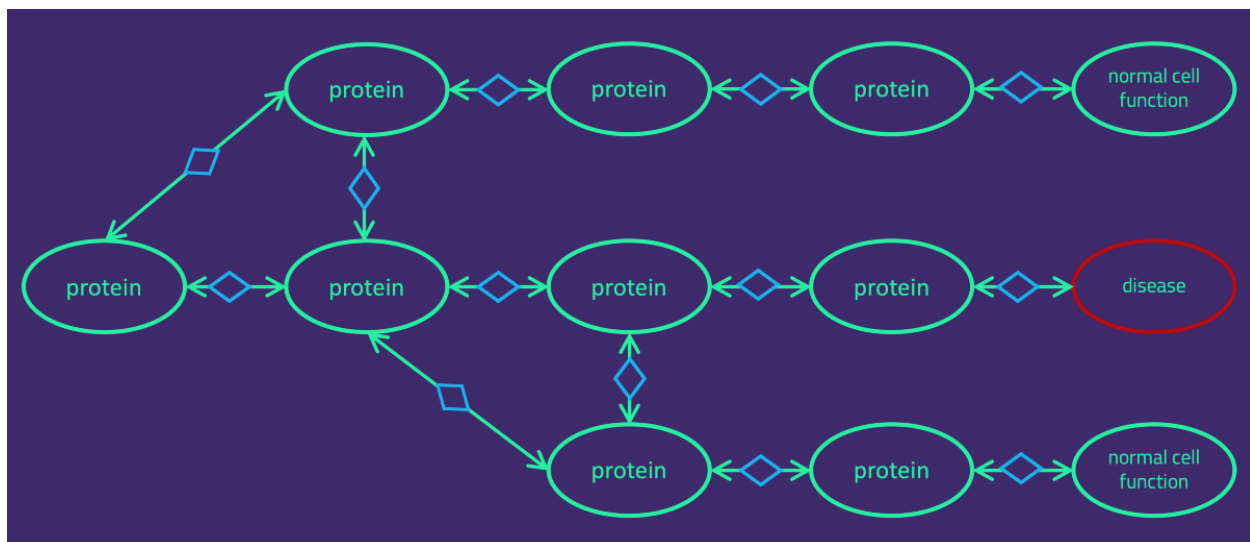
```
target-identification-rule sub rule,
when {
  $d isa disease;
  $p1 isa protein; $p2 isa protein;
  ($d, $p1) isa protein-disease-association;
  ($p1, $p2) isa protein-protein-association;
}, then {
  ($d, $p2) isa potential-target-identification;
};
```

The syntax for this rule can be seen on the left, where we are now using the protein which is directly related to the disease and the proteins which are related the directly related protein. A visual representation of this rule is as follows:



The dotted relations on the top are the inferred protein-protein-association relations which were created due to the logic of the first rule, while the ones on the bottom are the inferred protein-target-identification relations created by the second rule.

As much as we want it to be, Biology is not as simple as assumed above. Pathways are not linear in nature but reveal much more complex sets of relations. Proteins that take part in disease pathways are also responsible for causing a variety of normal cell functions. A slightly more realistic model would look like as follows:



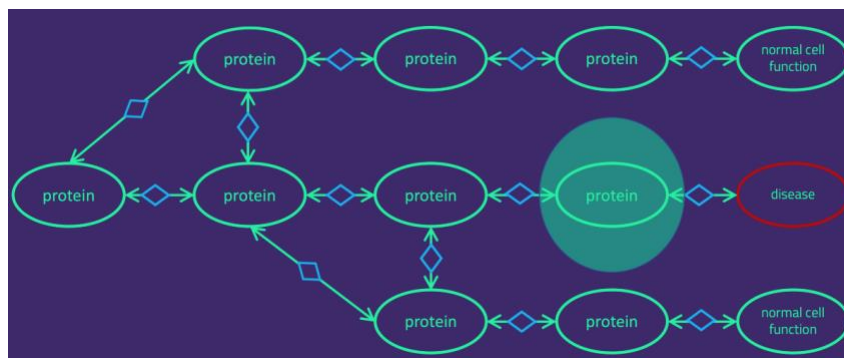


*Question: Given this model, what are the potential targets for the disease Melanoma that do not take part in any normal cell function pathways?*

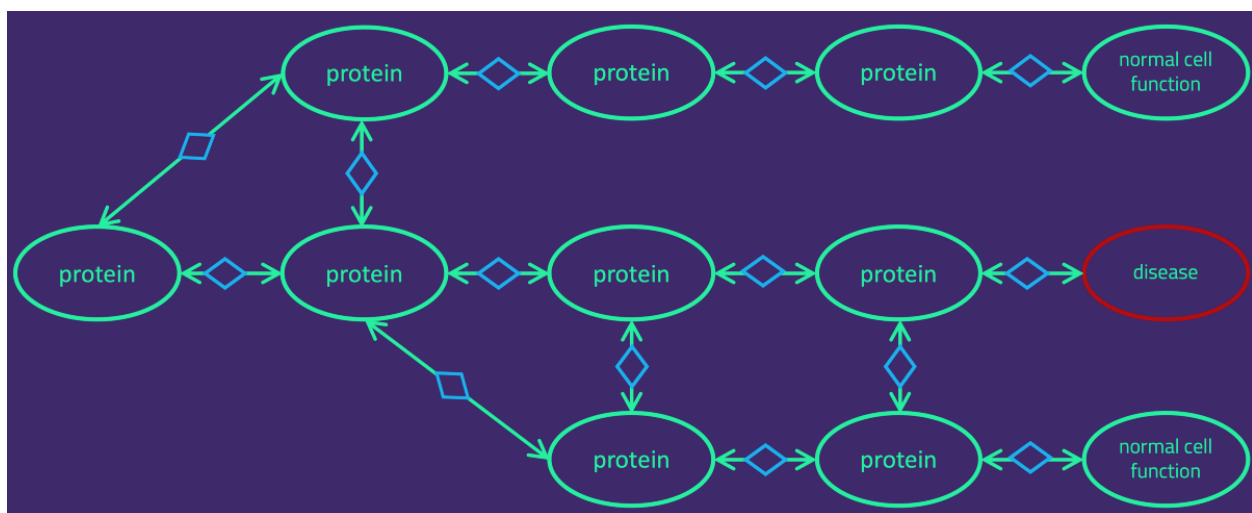
```
match
  $d isa disease, has name 'melanoma';
  $p isa protein;
  $cf isa normal-cell-function;
  $r ($d, $p);
  not { $r2($p, $cf); }
get;
```

This question can be translated into Graql with the help of negation, which enables us to ask for concepts which do not adhere to certain statements and in this case, enables us to retrieve the proteins that are not related to any normal cell function.

Answer: What is returned is the identification of the protein which is not causing any normal cell function.



But what if all proteins are responsible for cell functions? Can we get some sort of measure that can guide us to finding out an insight which may help our research?

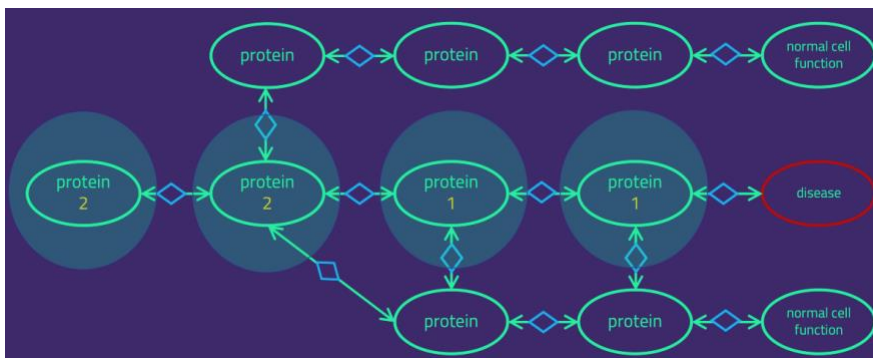


*Question: What are the potential targets for the disease Melanoma and what are the number of occurrences of those proteins in normal cell functions?*

```
match
  $d isa disease, has name 'melanoma';
  $p isa protein;
  $cf isa cell-function;
  $r ($d, $p);
  $r2($p, $cf);
get; group $p; count;
```

[Distributed analytics](#) is a set of scalable algorithms that allows you to perform computation over large amounts of data in a distributed fashion. The Graql query on the left shows an example of this which helps us answering our question.

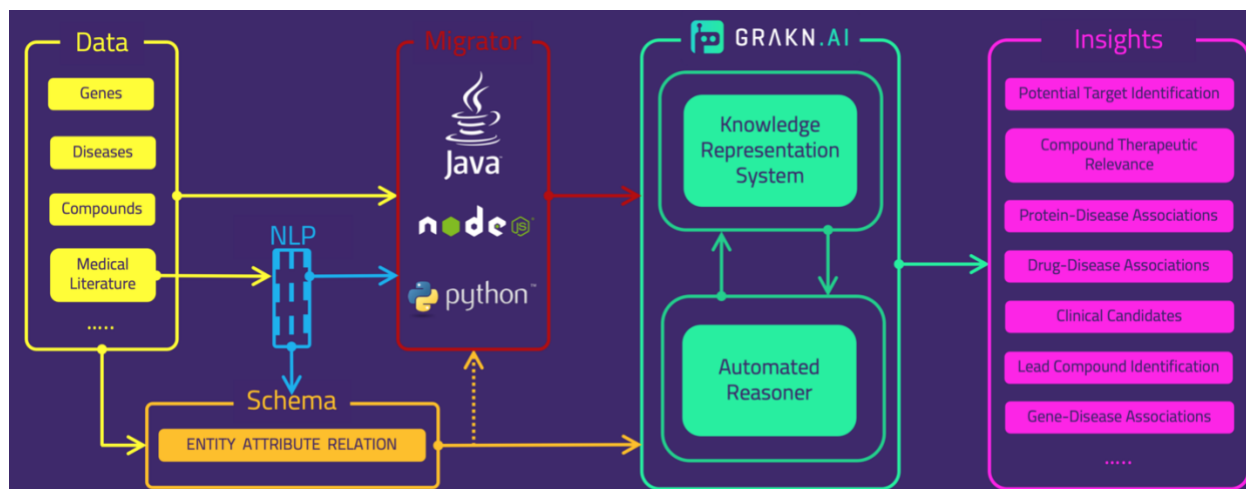
Answer:



Here, we get back exactly what we asked for — allowing us to shortlist the target proteins that we want to prioritise in our research; eventually expediting our research to viable compounds capable of treating a disease with minimal effects to your normal physiology.

### How do All the Pieces Fit Together in One Architecture


Now, let us take a step back, and look at how all of the components of building a drug discovery knowledge graph piece together.



We start with the data which can come from multiple sources and in various formats. That raw data is used to create a [schema](#) (high level data model) to enforce a structure on the raw data. Once that is done, we use one of [Grakn's clients](#) to migrate the instances of data into Grakn making sure every insertion adheres to the schema. Grakn stores this in its knowledge representation system, which can be queried for insights to discover complex relationships or even test out hypotheses. These insights can already be in the graph or even be created at the time of query by the reasoning engine.

## Conclusion

So, we know that Data Driven Drug Discovery is extremely promising in revolutionising global health care. We understand that there are barriers and bottle-necks associated with achieving it. I hope to have shown that Grakn can help to bring us many steps closer to efficiently, effectively and economically discovering drugs which and cure and treat those patients with diseases whose medications are still at large. In summary, Grakn helps to solve the three key challenges in drug discovery when it comes to handling data:

 <b>GRAKN.AI</b>		
Integration	Normalisation	Discovery
Ingest and <b>integrate</b> complex networks of <b>biological</b> data into one collection– a <b>knowledge graph</b>	Impose an <b>explicit structure</b> to <b>contextualise</b> the <b>relations</b> between <b>compound, gene</b> and <b>disease</b> networks	Use <b>automated reasoning</b> and <b>analytics</b> to discover and interpret <b>potential targets</b> and <b>drug candidates</b>



For more information, please visit [grakn.ai](https://grakn.ai) or contact us at [enterprise@grakn.ai](mailto:enterprise@grakn.ai).