Stemaway Presentation

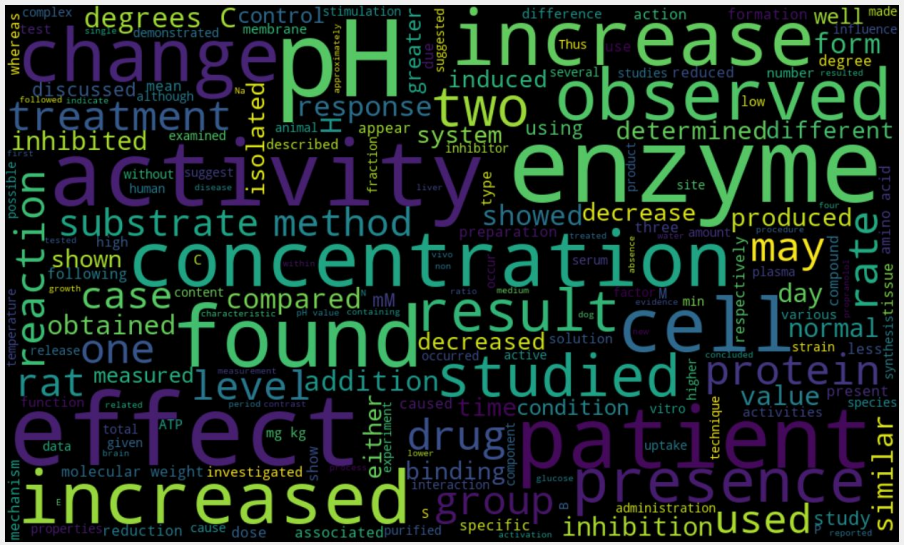
Team 7

Team Members

1. Charles Im
2. Yumin Guo
3. Akansha

Data Gathering

- Web scrape for Pubmed article archives
- Extracted abstract into data frames
- Use csv files for storage
- Obtained drug list from Drug Bank Online
- Obtained gene list from PharmGKB



Preprocessing

- **Removed a list of common words** from the genes (e.g. “cat”, “was” …)
- **Removed the dots (periods) within sentences** (used for abbreviation), without changing the performance of the sentences to be parsed.
- Looked for sentences that **contain exactly one drug name and one gene name**; then parsed those sentences

Dependency Parser

- What is it?
- A dependency parser tries to figure out the **grammatical structure of sentences**.
- It focuses on **how words are dependent** on each other (i.e. Which words are correlated and how?)

- Tool used: **Stanford Parser** (Jython interface)

E.G.

"Haloperidol (1 mg/kg) decreased the apparent Km of striatal TH for the pteridine cofactor."

'**nsubj [nominal subject]**
(decreased-6, Haloperidol-1)',
'**num [numeric modifier]**
(mg/kg-4, 1-3)',
'**appos [appositional modifier]**
(Haloperidol-1, mg/kg-4)',
'**root**
(ROOT-0, decreased-6)',
'**det [determiner]**
(Km-9, the-7)',
...

Stanford Parser

- With the drug and gene of each sentence, find the **shortest path** connecting (from) the drug and (to) the gene.
- Turn the results into a **large dependency matrix**, (drug-gene pairs as rows and relationships as columns), then pass to the EBC algorithm.

E.G.

"Haloperidol (1 mg/kg) decreased the apparent Km of striatal TH for the pteridine cofactor."

Drug: *Haloperidol*
Gene: *TH*
Relationship:
['nsubj', 'decreased', 'dobj', 'Km', 'prep_of']

Drug_Gene	['advmod', 'exhibits', 'nsubj', 'activity', 'appos']	['amod', 'acetyltransferase', 'conj', 'decarboxylase', 'conj', 'acetylcholinesterase', 'appos']	['amod', 'activity', 'prep_in', 'exhibited', 'prep_without', 'factor', 'appos']
tyrosine_TH	1	0	0
choline_AChE	0	1	0
tyrosine_NGF	0	0	1
morphine_TAT	0	0	0
acetate_PAH	0	0	0

Problems with the Current Parser

E.G.

*"A comparative study of fluorescences parameters of **HDC** and its inhibitory complexes with methyl ester of **histidine** (MEH), hydroxylamine and p-chloromercuriumbensoate is carried out."*

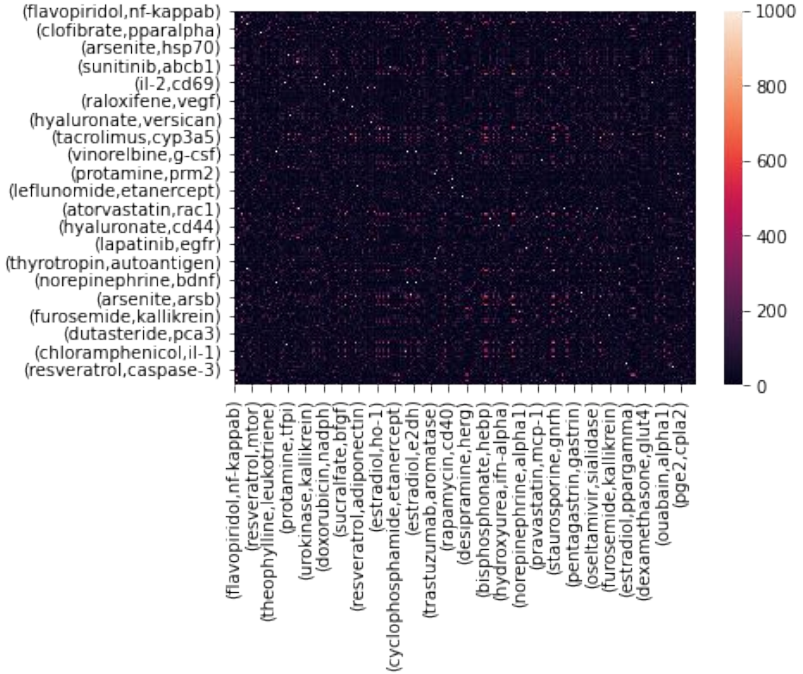
→ ["of"]

- The parser sometimes connects words that don't seem to be related.
- Some drug-gene relationships are too short to be useful.
  (we removed the relationship if the pair is connected by 0 or 1 word)

- The parser itself might be **making some mistakes consistently**.
- This version of parser treats every single word separately.

EBC (Unsupervised Portion)

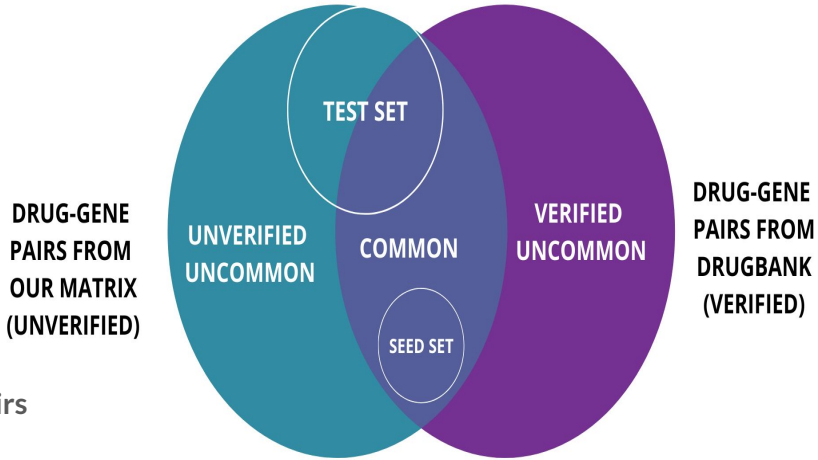
- Biclustered drug-pairs in the dependency matrix
- 1000 iterations used
- Aggregate the interactions for co-occurrence matrix



EBC (Supervised Portion)

Inputs:

1. TEST SET:
   - List of size 100
     - 50 pairs from DrugBank (Verified)
     - 50 pairs from our matrix(Unverified)
2. SEED SET:
   - List of size 1,2,3,4,5,10,100
     - All the pairs are verified DrugBank pairs



Scoring

Scoring Function

```
#Scoring function
scores = {}
for t in test_set:
  val = []
  ranks = {}
  #if the pair in test_set is in co-occurence matrix
  if t in DgPairs:
    #then sort the df on the basis of the column corresponding to the pair
    sorted_mat = df3.sort_values(by = t, ascending = False)
    rank = 1
    #Assigning ranks to all the drug-gene pair on the basis of their position
    for drugGene in sorted_mat['Drug Gene']:
      ranks[drugGene] = rank
      rank += 1
```

Sorted Matrix on the basis of flavopiridol,nf-kappab

	Drug Gene	(flavopiridol,nf-kappab)	(tnf-2,tnf-r1)	(il-2,il-5)	(il-11,il-10)	(fgf-7,fgf-2)	(clopidogrel,p-selectin)	(fgf-7,fgf-1)	(il-11,il-13)
0	(flavopiridol,nf-kappab)	1000	17	6	3	6	10	8	11
3071	(theophylline,nf-kappab)	1000	17	6	3	6	10	8	11
296	(menadione,egfr)	1000	17	6	3	6	10	8	11
729	(forskolin,cftr)	1000	17	6	3	6	10	8	11
3116	(rapamycin,gcn2)	547	16	18	6	10	12	9	12

Scoring

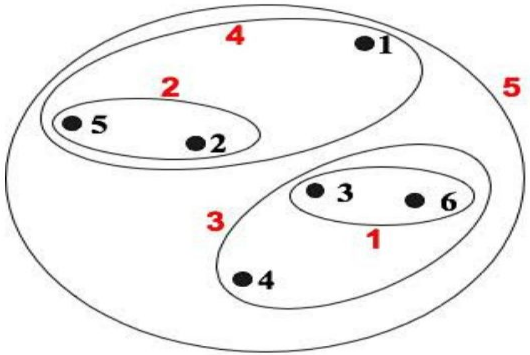
	Drug, Gene	Scores
0	(imiquimod,tlr7)	12859
1	(pgi2,ptgis)	20853
2	(warfarin,vkorc1)	20322
3	(gefitinib,egfr)	7026
4	(ezetimibe,npc1l1)	7901
5	(g-csf,csf3r)	8330
6	(doxorubicin,top2a)	10576
7	(aripiprazole,drd2)	13594
8	(vasopressin,avpr1b)	16728
9	(cerulenin,fasn)	14211

First 10 Scores/ Ranksum of Seed Set size 10

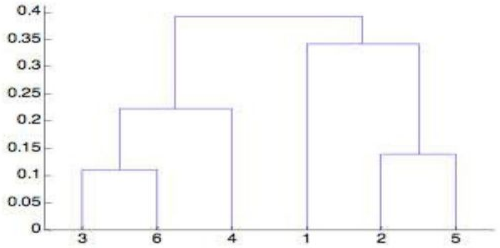
```
#Taking the ranksum that correspond to the seed_set members
for DG in sorted_mat['Drug Gene']:
  if DG in seed_set:
    val.append(ranks[DG])
    #The ranksum are the scores
scores[t] = sum(val)
```

Scoring Function

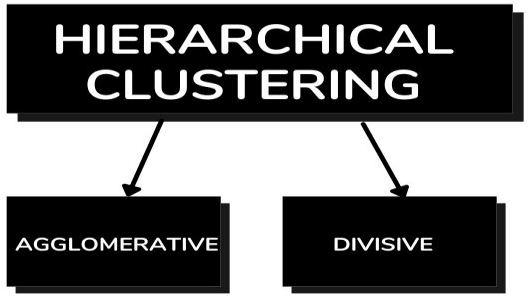
Dendrogram



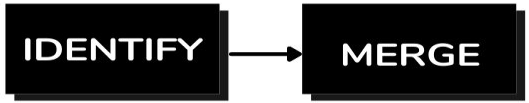
Nested Clusters



Dendrogram



What is a Dendrogram?



Implementation

1

CORRELATION

- Co-occurrence matrix -> correlation matrix
- ρij -> co-efficient of correlation

2

HIERARCHICAL CLUSTERING

- Correlation matrix -> distance matrix
- 1- ρ distance metric

3

CLUSTER ASSIGNMENT

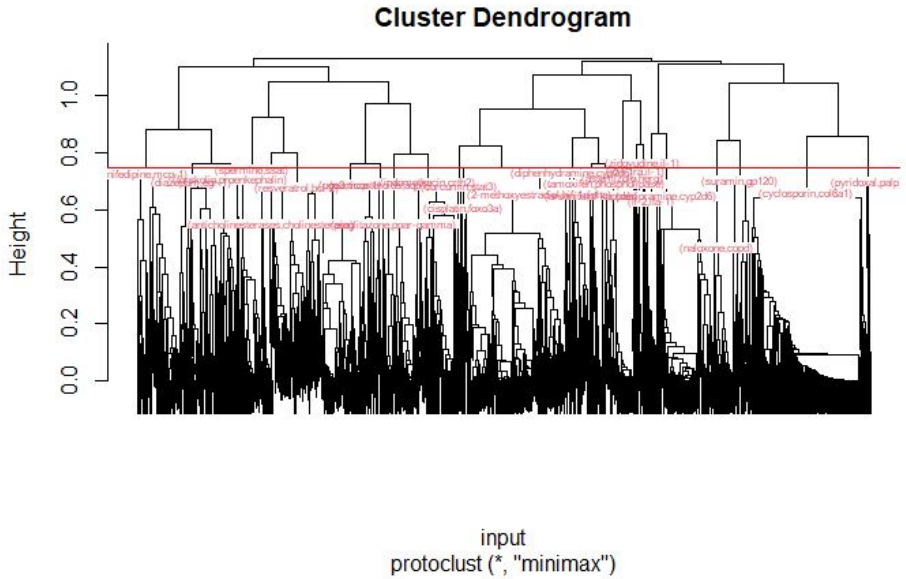
Cluster assignment by cuttting the dendrogram at a given height

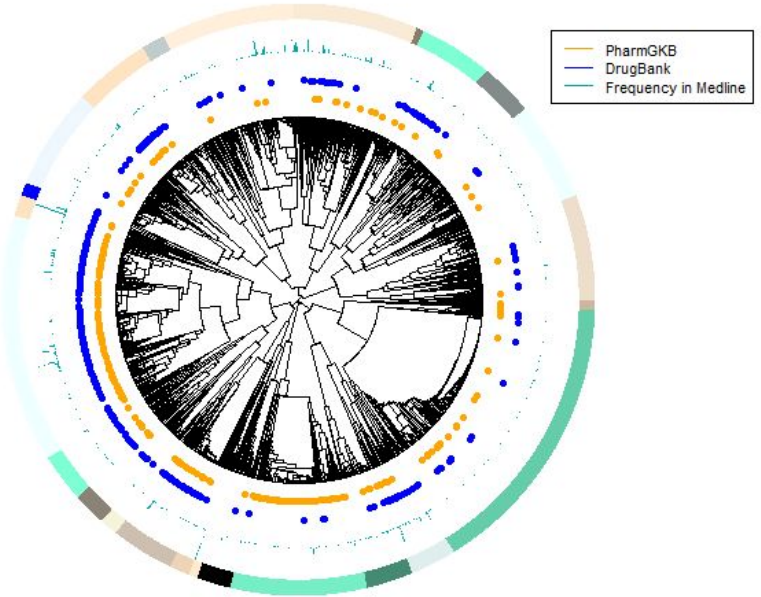
4

PLOT

- Cluster assignent
- tip markers for the existence in DrugBank and PharmGKB
- Bars accounting for frequency

Dendrogram Plots





Final Thoughts

- Charles
  - It was definitely a hard project but i believe that my team really held it together and pulled though
- Yumin
  - Big thanks to all the resources and our great teammates!
- Akansha
  - I am really thankful to the STEM- Away team for organising this and to my wonderful teammates for always being there when help was needed.