An analysis of the cost-effectiveness of research into candidate Prostate Cancer drugs determined by GenoPredict

B. Liu, N. V. Murolo, S. Vazquez, E. F. Yang

MTF Team: Whitney Young Advanced Thinkers

Abstract— **Prostate Cancer (PC) is the second most common form of cancer among men, as well as the second most deadly. Despite high incidence and mortality rates, little is known about the cause of the disease; late stages of PC are considered incurable. In this publication, we analyze potential treatments (identified by GenoPredict, an implementation of a drug repositioning algorithm) and their implications. To do so, we begin by providing the methodology by which possible treatments were procured, as well as an interpretation of the statistical significance of said methods. Particularly, we discuss the applications of network theory, and how GenoPredict is able to generate the enrichment scores via pathway analysis algorithms. We then relate the shortcomings of the use of heuristics in these algorithms to the sometimes inaccurate results of GenoPredict. With these ideas in mind, we are able to perform an informed cost-benefit analysis of low-scoring treatment options. We disqualify solutions that are predicted to be statistically efficient, but under more careful scrutiny are revealed to be a matter of chance. The study concludes with a final presentation of the most efficient solution among those considered, which happens to be Lorazepam, commonly known as Ativan, an anti-anxiety medication. We follow with a data-driven analysis of how its introduction into the PC drug treatment market might result in a nearly $3 billion total cost reduction for consumers at the end of a projected 5-year period.**

I. Technology Overview

In the search for treatment of new diseases, it is commonplace to "reposition," or repurpose, existing drugs, to see if they are effective. GenoPredict is an algorithm that was created to help us do just that-identify possible drugs for repositioning [1]. GenoPredict was the starting point in our search for a better treatment for Prostate Cancer, and, therefore, is a very fitting place to begin our publication. It is key we understand the functionality of GenoPredict, in order to successfully draw conclusions from the data it provides us with. We will first summarize the steps Wang and Xu took to design GenoPredict, then provide an in-depth explanation of each step. 1) A Genetic Disease Network (GDN) was constructed to display the overlap between genes in various diseases. 2) A network-based algorithm was applied to the GDN in order to find diseases similar to PC. 3) A function was created to identify the most probable treatments for diseases similar to PC, and thus, PC itself.

A. Genetic Disease Network

Wang and Xu created their GDN using data from the Catalog of Published Genome-Wide Association Studies. This source provided a master list of known disease-gene pairs, as well as other relevant biological information.

The strength of relations between items in the dataset, or "edges," were defined using the cosine similarity coefficients of disease-associated genes [2].

B. PageRank, the network-based algorithm

Next, Wang and Xu needed to identify the diseases that shared gene identifiers with PC. To do so, they applied the PageRank algorithm, an algorithm originally used by search engines to display the most relevant search results [3].

Like a series of web pages, with links both leading to and away from any given page, the body's health is majorly dictated by a series of protein reaction chains. These chains similarly have detectors "in," and output proteins "out." We first want to find a list of all inputs majorly affecting a particular set of output genes, namely, the same genes that have been found to correlate with Prostate Cancer; then, we want to identify which of these inputs are also related to diseases similar to PC. Additionally, it is important to note that not all connections are made equal. Some genetic links are stronger than others. The final relationship is expressed by the coefficient the algorithm outputs, which will be explained next.

Mathematically, we can represent this series of reaction chains as a square matrix, A, with elements corresponding to the edge strength between proteins. Picturing the links this way, the matrix acts as a method of transformation of any input protein to an output protein. We are searching for the strength of the link between genes indicative of PC and the occurrence of other related illnesses. In other words, we want to find the magnitude of a vector that will get us from the proteins indicative of one disease to the proteins indicative of PC. The length of this vector directly correlates to the strength of the edges, which tells us how related the input and output actually are. Since the solution is a steady-state probability vector, we know it to be one of the eigenvectors of A. Additionally, we know that the solution will be the dominant eigenvector of A, as we are definitely looking for the strongest relationship.

Thus, we can apply the power method of solving for the eigenvector. Repeatedly raising the exponent of A, then normalizing the matrix, will cause all areas of the vector field to approach the dominant eigenvector. The matrix is normalized to avoid receiving an arbitrary, non standardized output. Wang and Xu define the convergence of the aforementioned vectors as when the change in magnitude is smaller than 10-6. Conveniently, the output of the power method is the eigenvalue itself, so we need not compute more.

After retrieving these values, Wang and Xu compared them to the classification info given in the IDC10, which verified the accuracy of their algorithm.

C. The function in Question

Now that the related diseases were rallied up, it was necessary to analyze the effect of various drugs on the diseases as a whole. Using data from the GDN, we can identify which drugs treat which diseases.

To predict how well a drug will treat Prostate Cancer, we need only sum over the diseases it can treat, taking into account how related that disease is to PC.

$$\sum_{i=1}^{n} R_{\text{disease}\,i}$$

This leaves us with predicted rankings for how well various drugs will treat PC. Looking at the rankings to see how well the FDA approved drugs did, GenoPredict succeeded in identifying 25 of 27 drugs, placing the median of these in the 84th percentile. This seems to indicate that the algorithm worked well, as it can be assumed that the drugs approved for treatment of PC would outperform most other drugs [1].

However, we are also given many drugs that scored higher than FDA approved drugs. These are central to our research.

II. Data Methodology

We were able to obtain most of our data from government sources, like cancer.gov, and from the GenoPredict publication, in the form of CSVs, tables, and raw numbers. We used Jupyter, combined with the pandas, numpy, and fuzzywuzzy libraries to do most of our data processing.

From cancer.gov, we were able to obtain a massive dataset on prostate cancer patient prescriptions for patients on medicare part D. [4] In this dataset, prescriptions with less than 11 claims were marked with a "#" symbol in order to protect the privacy of the patients, and so we simply treated these prescriptions as if there were no claims for them, as they likely had a negligible enough market share for us to ignore. We refer to this dataset as our prescription dataset.

On the other hand, we were able to obtain both the list of GenoPredict rankings and a list of current FDA approved drugs for PC from the authors of the GenoPredict publication. [1] However, this list of FDA approved drugs used the formal names of many of the drugs, which resulted in many of the drugs not being findable at first in the prescription dataset. Because there were around 5,600 different prescription drugs in the prescription dataset, we resorted to more automatic methods in order to find close or correct names for the 27 different FDA approved drugs. To do this, we leveraged the fuzzywuzzy library to provide us with ranked searches of the formal names, and all of the brand names that we could find, and we selected the name with the highest claim to substitute back in and find the number of claims in the prescription dataset for each of the approved drugs. In this way, we were able to obtain the number of claims for each of these drugs, and in doing so, the equivalent market share.

```
In [510]: # try finding them from the original names
          for drug in approved:
              closest_names = process.extract(drug, list(df.index))
              print(drug, closest_names)

abarelix [('RALIX', 72), ('ANAGRELIDE HCL', 68), ('AVAR', 68), ('LASIX', 68), ('LIDEX', 68)]
abiraterone [('ESTRONE', 75), ('CERON', 72), ('WATER', 72), ('KERLONE', 69), ('IRON', 68)]
aminoglutethimide [('TESTIM', 75), ('OLUX', 68), ('ETHMOZINE', 64), ('MARINOL', 64), ('NICOMIDE', 64)]
bicalutamide [('BICALUTAMIDE', 100), ('UTA', 90), ('NILUTAMIDE', 82), ('TEBAMIDE', 77), ('FLUTAMIDE', 76)]
cabazitaxel [('AVITA', 72), ('ZIAC', 68), ('PACLITAXEL', 67), ('AZASITE', 64), ('BALZIVA', 64)]
capecitabine [('DECITABINE', 82), ('TOBI', 68), ('GEMCITABINE HCL', 67), ('AMBIEN', 66), ('BIAFINE', 66)]
chlorotrianisene [('MOTRIN', 75), ('CHLOROTHIAZIDE', 73), ('TIGAN', 72), ('TRIAZ', 72), ('ENEMA', 68)]
conjugated estrogens [('ESTROGEL', 79), ('ESTRONE', 77), ('DELESTROGEN', 74), ('GENEBS', 72), ('EDEX', 68)]
cyclophosphamide [('CYCLOPHOSPHAMIDE', 100), ('CHLORPROPAMIDE', 73), ('ARIMIDEX', 69), ('AXID', 68), ('CLEOCIN PHOSPHATE
', 67)]
degarelix [('RALIX', 72), ('LASIX', 68), ('LIDEX', 68), ('RELPAX', 66), ('RELURI', 66)]
diethylstilbestrol [('DIETHYLSTILBESTROL', 100), ('ESTROGEL', 77), ('DETROL', 75), ('ETHYOL', 75), ('ESTRADIOL', 72)]
docetaxel [('DOCETAXEL', 100), ('CEDAX', 72), ('CETRAXAL', 71), ('DENTAGEL', 71), ('PACLITAXEL', 63)]
enzalutamide [('UTA', 90), ('NILUTAMIDE', 82), ('TEBAMIDE', 77), ('FLUTAMIDE', 76), ('BICALUTAMIDE', 75)]
esterified estrogens [('ESTROGEL', 79), ('ESTRONE', 77), ('DELESTROGEN', 74), ('GENEBS', 72), ('NTS', 72)]
estradiol [('ESTRADIOL', 100), ('DEPO-ESTRADIOL', 90), ('ESTRADIOL VALERATE', 90), ('ETHINYL ESTRADIOL', 90), ('IPOL', 7
7)]
estramustine [('CHOLESTYRAMINE', 69), ('CHOLESTYRAMINE LIGHT', 68), ('MUSE', 68), ('LOMUSTINE', 67), ('STAVUDINE', 67)]
estrone [('ESTRONE', 100), ('ESTROGEL', 80), ('4-DIHYDROTESTOSTERONE', 77), ('ALDOSTERONE', 77), ('DELESTROGEN', 77)]
ethinyl estradiol [('ETHINYL ESTRADIOL', 100), ('ESTRADIOL', 90), ('DEPO-ESTRADIOL', 74), ('ADVIL', 72), ('OCL', 72)]
flutamide [('FLUTAMIDE', 100), ('UTA', 90), ('NILUTAMIDE', 84), ('TOLBUTAMIDE', 80), ('BICALUTAMIDE', 76)]
goserelin [('CEREFOLIN', 67), ('REGLAN', 66), ('RELURI', 66), ('VELTIN', 66), ('GAS RELIEF', 63)]
histrelin [('ASTELIN', 75), ('AHIST', 72), ('ACITRETIN', 67), ('DEHISTINE', 67), ('HYTRIN', 67)]
leuprolide [('LEUPROLIDE ACETATE', 90), ('LIDEX', 80), ('LUPRON', 75), ('LURIDE', 75), ('NEUPRO', 75)]
mitoxantrone [('MITOXANTRONE HCL', 95), ('MINTOX', 75), ('NOVANTRONE', 73), ('OXANDROLONE', 70), ('INTRON A', 68)]
nilutamide [('NILUTAMIDE', 100), ('FLUTAMIDE', 84), ('BICALUTAMIDE', 82), ('TOLBUTAMIDE', 76)]
radium 223 dichloride [('LURIDE', 75), ('FLUORIDE', 72), ('LOPID', 72), ('OVIDE', 72), ('SODIUM CHLORIDE', 72)]
sipuleucel-t [('ACCU-CHEK SAFE-T-PRO', 86), ('T.E.D. ANTI-EMBOLISM STOCKINGS', 86), ('URIMAR-T', 86), ('IPOL', 68), ('IB
U', 66)]
triptorelin [('RELI ON', 75), ('CHLORDIAZEPOXIDE-AMITRIPTYLINE', 68), ('IPOL', 68), ('PERPHENAZINE-AMITRIPTYLINE', 68),
('RITALIN', 67)]
```

name	trade names	number of claims	price for treatment	total treatment costs
bicalutamide	bicalutamide	15265	1649.0	2.5172e+07
enzalutamide	xtandi	4224	236000.0	9.96864e+08
abiraterone	zytiga	3527	115000.0	4.05605e+08
flutamide	flutamide	256	448.0	114688
nilutamide	nilutamide, nilandron	112	40000.0	4.48e+06
estradiol	estradiol	76	NaN	NaN
leuprolide	eligard	72	NaN	NaN
degarelix	firmagon	52	NaN	NaN
goserelin	zoladex	30	NaN	NaN
cyclophosphamide	cyclophosphamide	26	NaN	NaN
estramustine	emcyt	24	8788.0	210912
docetaxel	docetaxel	23	14000.0	322000
conjugated estrogens	premarin	18	NaN	NaN

III. Mathematics Methodology

We calculated estimates for the current drug treatments, simply using the number of claims in our dataset and the estimated prices for a full treatment to get the approximate current market shares of each drug. By doing so, we were also able to find out how much our dataset needed to be scaled to predict the global dataset of a market share of $7.9 billion in 2016. We made several assumptions at the this step:

- We assume that we have found around 75% of the actual market share in our sample dataset (likely 25% were missed)
- We assume that there will be around a 30% lorazepam adoption in a timeframe of around 5 years
- We assume that the exponential growth rate of 4.8% growth in the prostate cancer drug market will continue over the next five years.

IV. Research Results

The code to generate our results is included below:

```python
import pandas as pd
import numpy as np
# fuzz is used to compare TWO strings
from fuzzywuzzy import fuzz

# process is used to compare a string to MULTIPLE other strings
from fuzzywuzzy import process

df = pd.read_csv("https://healthcaredelivery.cancer.gov/seermedicare/aboutdata/pedsf.ndc.bn.partd.table.prostate.csv.txt", sep='\t'
# In order to ensure confidentiality cell sizes less than 11 are masked (#).

df.columns = df.loc[1]
df = df.drop([0,1])
df = df.drop(map(str, range(2007, 2016)), axis=1).drop(float('nan'), axis=1)
df["2016"].loc[2] = 360057
df.columns.name = None

# set all #s to essentially 0
def convert_entry_to_number(x):
    if x == "#":
        return 0
    else:
        return int(x)
convert_entry_number_vec = np.vectorize(convert_entry_to_number)
df["2016"] = convert_entry_number_vec(df["2016"])
df = df.set_index("Brand Name")
approved = list(pd.read_csv("http://nlp.case.edu/public/data/PC_GenoPredict/PC_drugs_FDA_approved.txt", header=None)[0])

# try finding them from the original names
for drug in approved:
    closest_names = process.extract(drug, list(df.index))
    print(drug, closest_names)

def lookup(name):
    print(process.extract(name, list(df.index)), df.loc[process.extract(name, list(df.index))[0][0]]["2016"])
lookup("novantrone")

# replace with closer names - ones that have the highest count in the dataset
approved_list = {"aberalix": [],
                 "abiraterone": ["zytiga"],
                 "aminoglutethimide": ["cytadren"],
                 "bicalutamide": ["bicalutamide"],
                 "cabazitaxel": ["jevtana"],
                 "capecitabine": ["xeloda"],
                 "chlorotrianisene": [],
                 "conjugated estrogens": ["premarin"],
                 "cyclophosphamide": ["cyclophosphamide"],
                 "degarelix": ["firmagon"],
                 "diethylstilbestrol": ["diethylstilbestrol"],
                 "docetaxel": ["docetaxel"],
                 "enzalutamide": ["xtandi"],
                 "esterified estrogens": ["menest"],
                 "estradiol": ["estradiol"],
                 "estramustine": ["emcyt"],
                 "estrone": ["estrone"],
                 "ethinyl estradiol": ["ethinyl estradiol"],
                 "flutamide": ["flutamide"],
                 "goserelin": ["zoladex"],
                 "histrelin": ["vantas"],
                 "leuprolide": ["eligard"],
                 "mitoxantrone": ["novantrone"],
                 "nilutamide": ["nilutamide", "nilandron"],
                 "radium 223 dichloride": [],
                 "sipuleucel-t": [],
                 "triptorelin": []}

usage_numbers = pd.DataFrame(columns=["name", "trade names", "number of claims"])
for drug, trade_names in approved_list.items():
    number_claims = sum([df.loc[process.extract(trade_name, list(df.index))[0][0]]["2016"] for trade_name in trade_names])
    tmp = pd.DataFrame({"name": drug, "trade names": ", ".join(trade_names), "number of claims": number_claims}, index=[0])
    usage_numbers = usage_numbers.append(tmp, ignore_index=True)

usage_numbers = usage_numbers[usage_numbers["trade names"] != ""]
usage_numbers = usage_numbers[usage_numbers["number of claims"] != 0]
usage_numbers = usage_numbers.sort_values("number of claims", ascending=False).set_index("name")

usage_numbers_with_prices = usage_numbers

usage_numbers_with_prices["price for treatment"] = np.nan
usage_numbers_with_prices["price for treatment"].loc["bicalutamide"] = 1649
usage_numbers_with_prices["price for treatment"].loc["abiraterone"] = 115000
usage_numbers_with_prices["price for treatment"].loc["enzalutamide"] = 236000
usage_numbers_with_prices["price for treatment"].loc["docetaxel"] = 14000
usage_numbers_with_prices["price for treatment"].loc["estramustine"] = 8788
usage_numbers_with_prices["price for treatment"].loc["flutamide"] = 448
usage_numbers_with_prices["price for treatment"].loc["nilutamide"] = 40000

# assume that we cover around 75% of the market
usage_numbers_with_prices["total treatment costs"] = usage_numbers_with_prices["number of claims"] * usage_numbers_with_prices["pri

total_sample_cost = usage_numbers_with_prices["total treatment costs"].sum()
total_sample_cost

total_sample_cost_adjusted = total_sample_cost/0.75
total_sample_cost_adjusted

current_global_cost = 7.9 * 10**9
current_global_cost

global_from_sample_ratio = current_global_cost / total_sample_cost_adjusted
global_from_sample_ratio

# assume around 30% market adoption by number of claims
lorazepam_market_adoption_percent = 0.3
predicted_usage_numbers_with_prices = usage_numbers_with_prices
predicted_lorazepam_adoption = predicted_usage_numbers_with_prices["number of claims"].sum() * lorazepam_market_adoption_percent
predicted_usage_numbers_with_prices["number of claims"] *= (1 - lorazepam_market_adoption_percent)
predicted_usage_numbers_with_prices = predicted_usage_numbers_with_prices.append(pd.DataFrame(dict(zip(predicted_usage_numbers_with
predicted_usage_numbers_with_prices["total treatment costs"] = predicted_usage_numbers_with_prices["number of claims"] * predicted_

total_sample_cost_lorazepam = predicted_usage_numbers_with_prices["total treatment costs"].sum()
total_sample_cost_lorazepam_adjusted = total_sample_cost_lorazepam / 0.75
total_sample_cost_lorazepam_adjusted

# 5 year projection
cost_percentage_gain = 1.048 ** 5
global_cost_projected = current_global_cost * cost_percentage_gain
global_cost_projected

global_cost_projected_lorazepam = total_sample_cost_lorazepam_adjusted * global_from_sample_ratio * cost_percentage_gain
global_cost_projected_lorazepam

cost_reduction_lorazepam = global_cost_projected - global_cost_projected_lorazepam
cost_reduction_lorazepam
```

In our results, we found an approximate $10 billion market for prostate cancer drugs, and an approximate $7.1 billion market for prostate cancer drugs when assuming that Lorazepam had around a 30% adoption rate in terms of number of claims, thus displacing other treatment methods. This means that with the addition of Lorazepam in the market, there is potential to have a nearly $2.9 billion decrease in total treatment costs for patients 5 years from today.

V. Conclusions and Recommendations

In conclusion, while the PageRank algorithm is very useful in the derivation of treatment rankings, we must also keep in mind the downsides of its application to the issue of medicine. For example, the aforementioned edge values are largely unknown in the field of biology; these values rely mainly on scientific consensus as opposed to concrete data. Minor fluctuation in these edge values could greatly sway the course of the algorithm. This is not necessarily a downside, but depending on the accuracy of current scientific understanding, it could prove these results incorrect. However, new information would simply allow the algorithm to be re-evaluated under new operating conditions, in order to return a more accurate result.

These considerations could explain the representation of only 25 of 27 of the FDA approved drugs in the prediction. Another plausible explanation for this is the mere lack of data. Again, this is not necessarily bad: it provides an incentive for improvement.

Finally, we must consider the problem of the dominant eigenvector. In a purely statistical world, there is nothing wrong with this approach to the problem. The strongest link is, quantifiably, the best. However, as much remains unknown in the fields of biology and medicine, this answer is not as certain. There may be a pathway through the protein reaction chains that demonstrates a statistically low correlation, but in effect, has the most impact. The possibility of invisible side-interactions going on is simply too high to ignore. Although this does not necessarily reduce the validity of any results of the study, it means we might be missing the bigger picture, something necessary for the eventual cure of Prostate Cancer. Overall, more medical testing and clinical trials are required for better results, but the saving an estimated three billion USD is a substantial amount for patients.

References

[1] X. R. Wang Q, "Drug repositioning for prostate cancer: using a data-driven approach to gain new insights.." https://www.ncbi.nlm.nih.gov/pubmed/29854243.

[2] J. P. Jiawei Han, Micheline Kamber, "Data Mining Concepts and Techniques."

[3] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web."

[4] N. C. Institute, "Frequency of prescription drugs on part d event file by brand drug name."

[5] M. V. F. L. C. A. G. A. Alice Dragomir, Daniela Dinea, "Drug costs in the management of metastatic castration-resistant prostate cancer in Canada." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4099156/.

[6] FDA, "ZYTIGA® (abiraterone acetate) Tablets ." https://www.accessdata.fda.gov/drugsatfda_docs/label/2015/202379s021lbl.pdf.

[7] R. Preidt, "A light breakfast might cut cost of pricey prostate cancer drug."

[8] F. G. M. T. B. Juan Hinojosa, Mauricio Mora Pineda, "Low-dose diethylstilbestrol (des) as frontline treatment for hormone-sensitive metastatic prostate cancer (pc) in a low resource setting.." http://ascopubs.org/doi/abs/10.1200/JCO.2016.34.15_suppl.e16516.

[9] D. G. T. W. F. Julia Clemons, L. Michael Glodé, "Low-dose diethylstilbestrol for the treatment of advanced prostate cancer." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3229833/.

[10] T. Hagen, "Enzalutamide pricing debate goes to washington."

[11] Xtandi, "Xtandi® (enzalutamide) capsules dosing and administration."

[12] R. T. R. P. G. N. K. L. A. B. S. P. R. R. R Collins, E Fenwick, "A systematic review and economic model of the clinical effectiveness and cost-effectiveness of docetaxel in combination with prednisone or prednisolone for the treatment of hormone-refractory metastatic prostate cancer." https://www.ncbi.nlm.nih.gov/books/NBK62261/.

[13] C. T. Advisor, "Prostate cancer treatment regimens."

[14] D. M. Caroline M. Perry, "Estramustine phosphate sodium." https://link.springer.com/article/10.2165/00002512-199507010-00006.

[15] PDR, "estramustine phosphate sodium - drug summary."

[16] A. V. P. R. R. B. K. J. M. B. R. P. Arlee Fafalios, Ardavan Akhavan, "Translocator receptor blockade reduces prostate tumor growth." https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2779999/.

[17] PDR, "flutamide - drug summary."

[18] PDR, "nilutamide - drug summary."

[19] G. V. Research, "Prostate cancer therapeutics market analysis by drugs (zytiga, gonax, lupron, zoladex, decapeptyl, eligard, vantas, casodex, xtandi, taxotere, jevtana, provenge, xofigo), by region, and segment forecasts, 2018 - 2025." https://www.grandviewresearch.com/industry-analysis/prostate-cancer-therapeutics-market.