

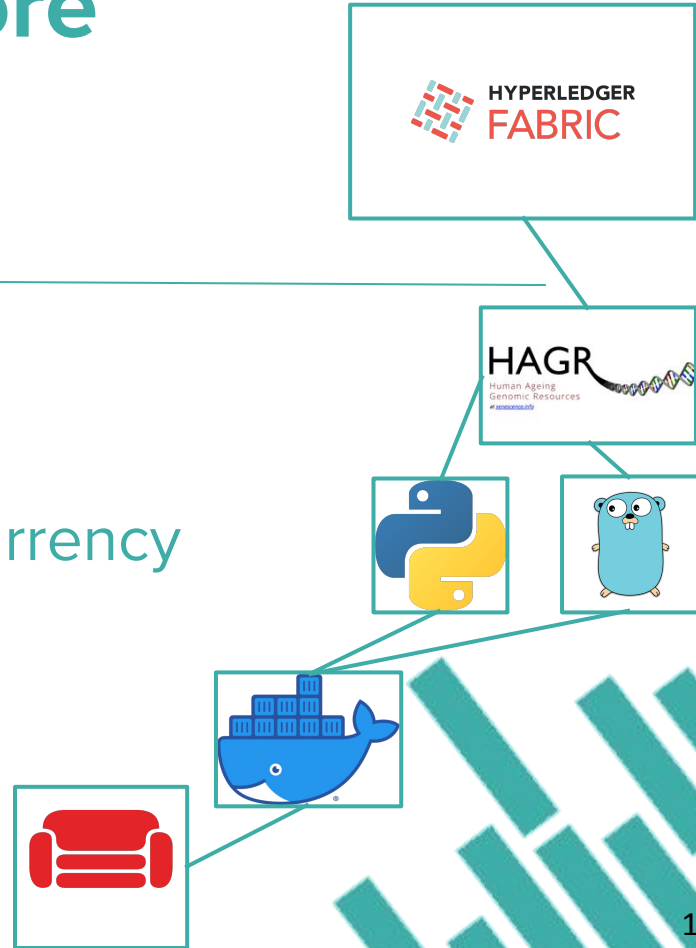
HyperLedger Fabric to store and share Human Ageing Genomics data

IBC Project - CSE 528 May 2021
Course Faculty - Dr. Donghoon Chang
Introduction to Blockchain and Cryptocurrency

MT19213
Ghanendra Singh
MTech Computational Biology

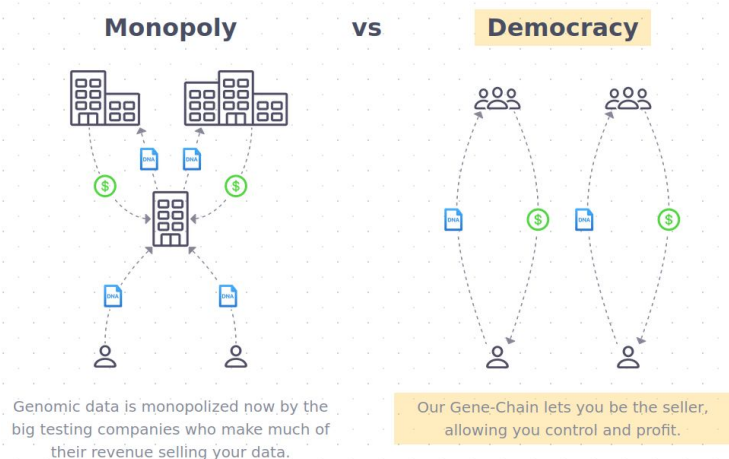


INDRAPRASTHA INSTITUTE of
INFORMATION TECHNOLOGY DELHI



- Blockchain Startups in Genomics

- EncrypGen develops software for genomic data, empowering patients and donors, facilitating health, business, and science.



Buy DNA tokens to buy DNA data

Bitcoin (BTC), DNA tokens, and soon other cryptocurrencies may be used to purchase genomic data, and other DNA products and health related services in the Gene-Chain Marketplace.

DNA tokens are an ERC-20 based cryptocurrency available on numerous crypto exchanges all over the world and exchangeable for Bitcoin (BTC), Ethereum (ETH) and several other altcoins.



How it works

If you have had your DNA tested you may upload your raw DNA data file and create a Gene-Chain profile now.

EncrypGen de-identifies the raw DNA data file by stripping it away from name, email, and other sensitive information. DNA data buyers search Gene-Chain profiles suitable for their projects and purchase de-identified genomic data with DNA tokens.

Coinmarketcap and Crypto exchange



EncrypGen DNA ☆

Rank #4045 Token On 791 watchlists

EncrypGen Price (DNA) **\$0.05402** ▼ 6.30%

0.0000009286 BTC ▼ 4.50%
0.00001363 ETH ▼ 15.47%

Low: \$0.0539 High: \$0.0587 24h

www.encyrpgen.com Explorers Community

Whitepaper

Contracts:
Ethereum: 0xef63...CdC572C

Tags: Health AI & Big Data Research View all

Market Cap \$3,832,203 ▲ 0.00%

Fully Diluted Market Cap \$3,831,872 ▼ 6.30%

Volume 24h \$1,309 ▲ 96.84%

Self Reported Circulating Supply 70,938,084 DNA

Volume / Market Cap --

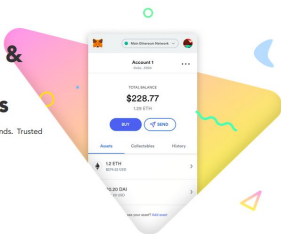
Max Supply 70,938,084
Total Supply 70,938,084

METAMASK Features Support About Build Download

A crypto wallet & gateway to blockchain apps

Start exploring blockchain applications in seconds. Trusted by over 1 million users worldwide.

Download now



Ref: <https://metamask.io/>

Ref: <https://coinmetro.com/>

CoinMetro
MOVING CRYPTO FORWARD

Markets Exchange

Way Beyond the Best Crypto Exchange

Buy, sell, & invest crypto. Be a part of the future of finance.

Get Started ★★★★★ See our reviews

Ref: https://coinmarketcap.com/currencies/encrypgen/?utm_medium=widget&utm_campaign=cmwidget&utm_source=encyrpgen.com&utm_content=encyrpgen

- Zenome plans to build a decentralized storage system for genomic data provided by network participants and supported financially with the help of internal cryptocurrency.



Decentralized
peer-to-peer network
constructed

for the secure trading of genomic data

Welcome to Zenome

A first decentralized Internet of genomic data and services

ZENOME MANUSCRIPT

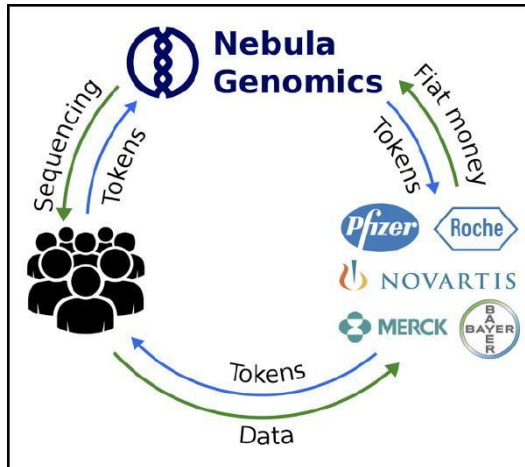
GENETIC TEST

The first genetic test with
continuous feedback on

blockchain technology from the Zenome platform

- We use Whole Genome Sequencing to decode 100% of your DNA and produce 10,000 times more data than other DNA tests like 23andMe and AncestryDNA.

Token flow



Human Art Project

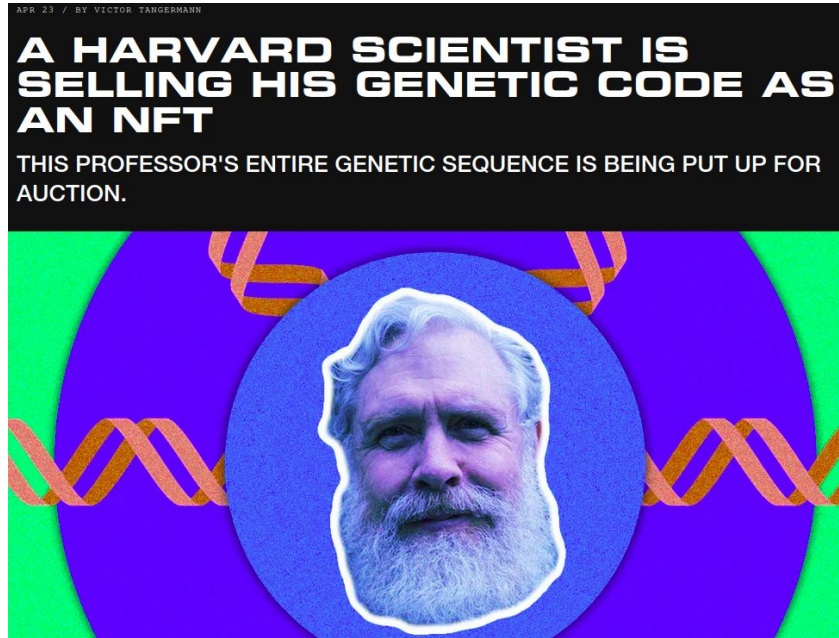
The Human Art Project uses next generation DNA Sequencing powered by Nebula Genomics to create personalized artwork based on your genome.

Choose between 5 unique styles to generate a piece of art that only YOU could create.

Nebula Genomics



- 23rd Apr 2021



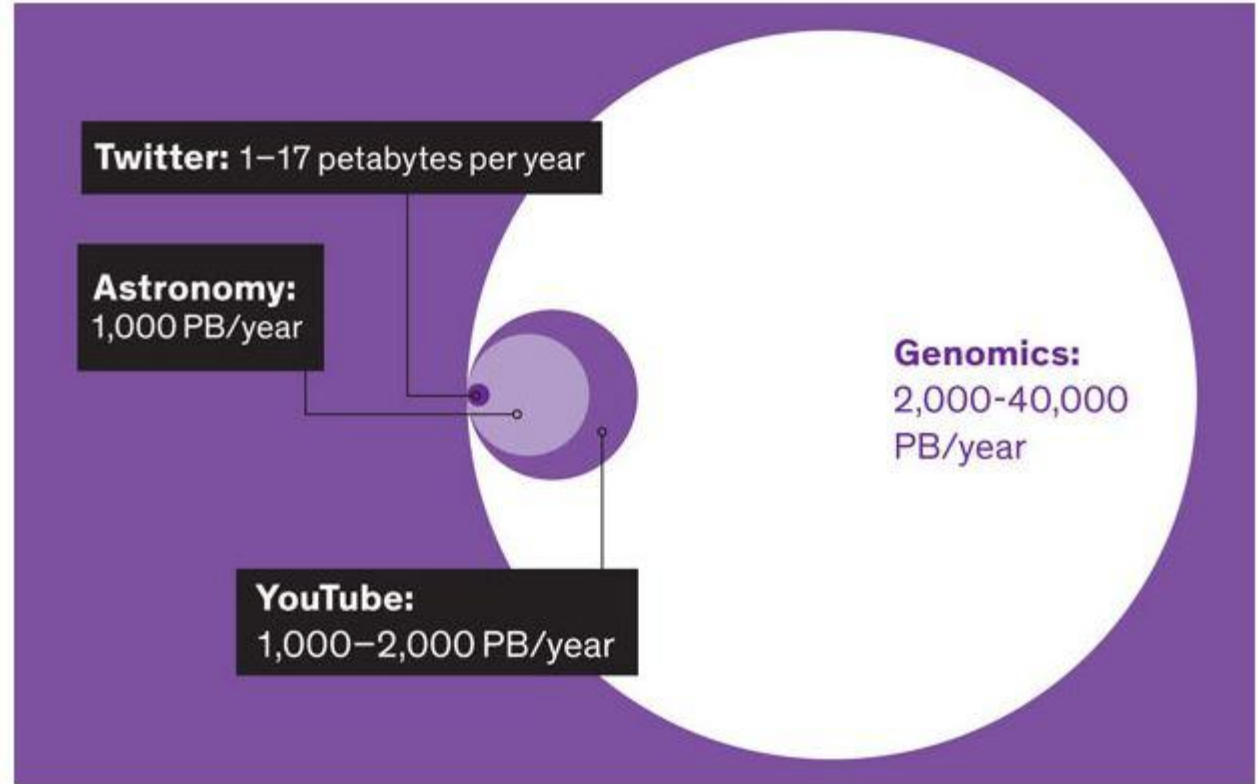
Ref:

<https://www.the-scientist.com/news-opinion/q-a-george-church-s-genome-up-for-auction-68682>

Ref: <https://futurism.com/neoscope/george-church-harvard-scientist-genetic-code-nft>

Projected annual storage in 2025

- Why secure Genomics Data Storage Needed ?



Source: ["Big Data: Astronomical or Genomical?"](#) PLoS Biology, 7 July 2015.

Research Paper



Technical Advance | [Open Access](#) | Published: 01 June 2020

Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts

[Gamze Gürsoy](#), [Charlotte M. Brannon](#) & [Mark Gerstein](#) ✉

[BMC Medical Genomics](#) **13**, Article number: 74 (2020) | [Cite this article](#)

9432 Accesses | 1 Citations | 9 Altmetric | [Metrics](#)

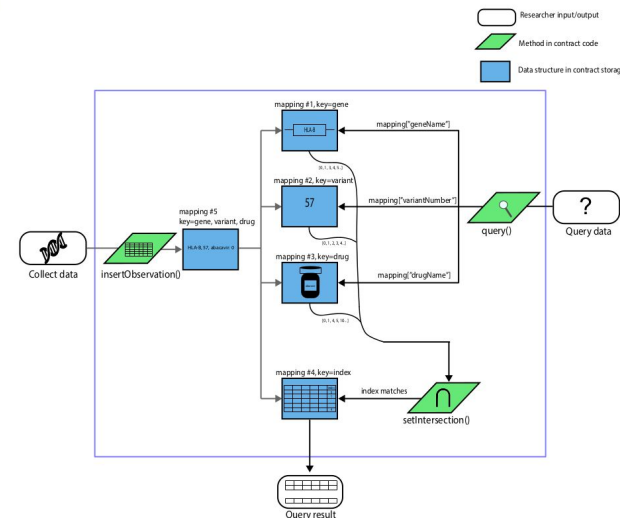


Figure from Paper

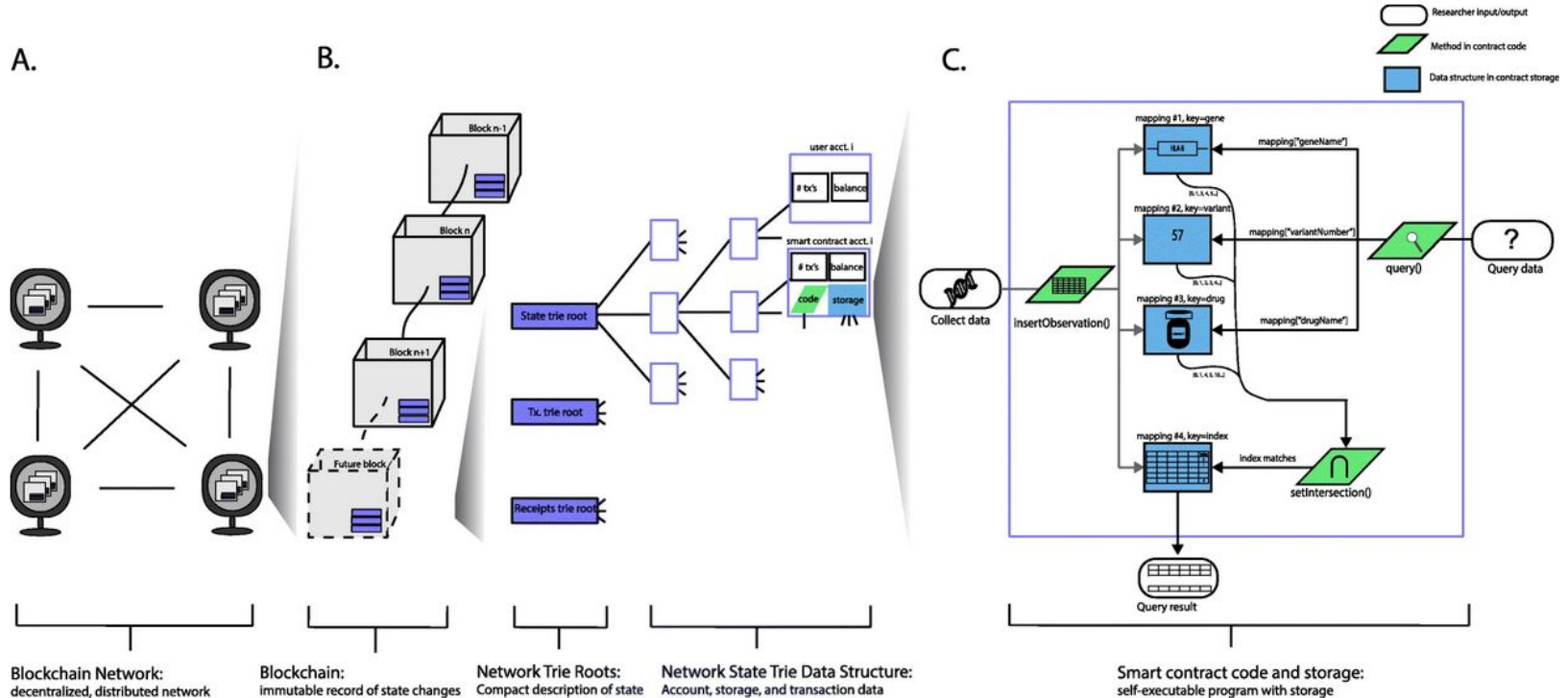
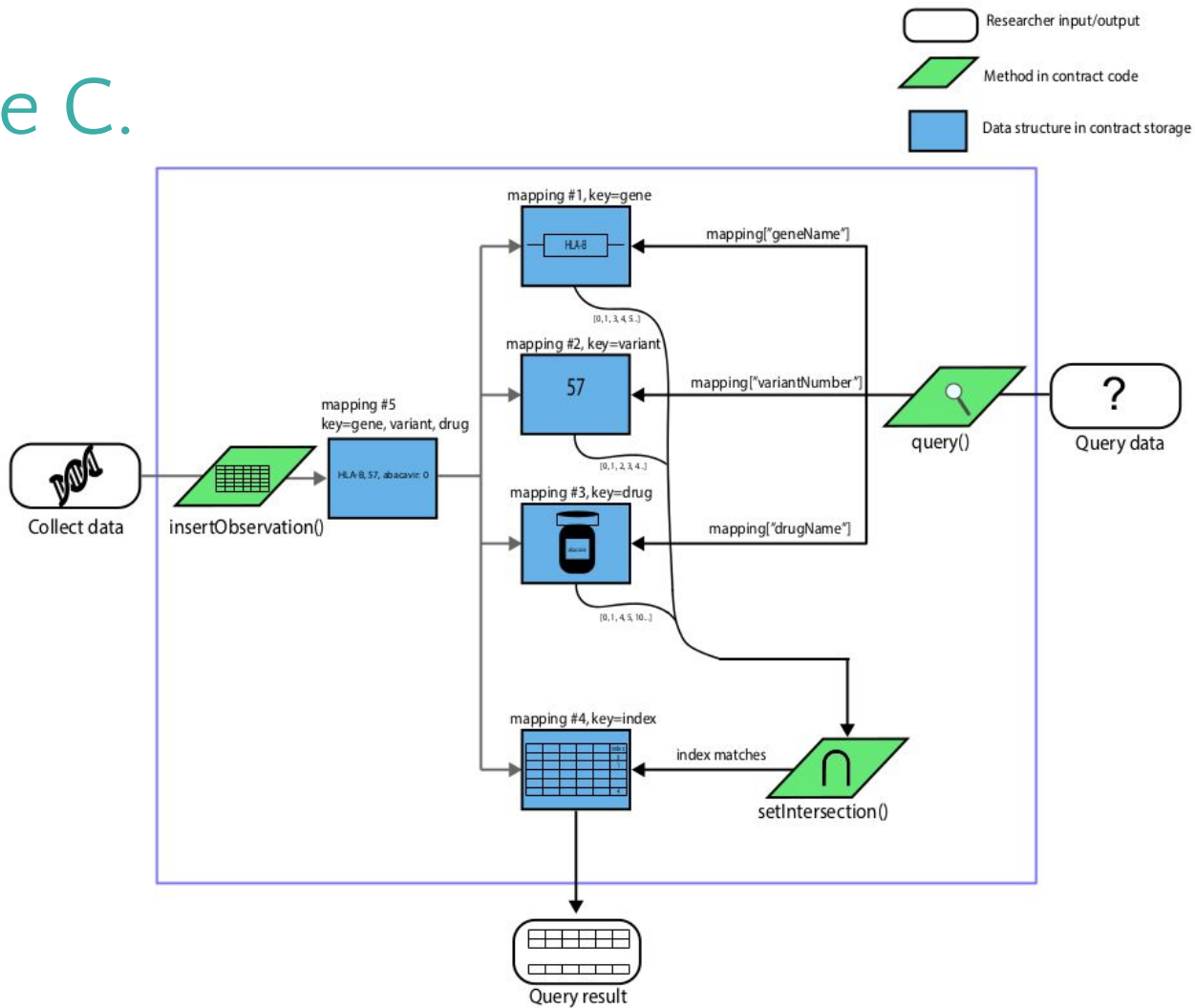


Figure C.



Algorithms

Algorithm 1 Challenge Solution - Insertion

```
1: procedure INSERT(STRING gene, UINT variant,  
   STRING drug, STRING outcome, BOOL relation, BOOL  
   sideEffect)  
2:   entry.push(gene, variant, drug, outcome, relation,  
   sideEffect)  
3:   if entry exists in database then  
4:     uniqueEntries[i].push(gene, variant, drug)  
5:     geneMapping[gene].push(ID)  
6:     variantMapping[variant].push(ID)  
7:     drugMapping[drug].push(ID)  
8:     database[ID].push(entry)  
9:     ID++
```

Algorithm 2 Challenge Solution - Query

```
1: procedure QUERY(STRING gene, STRING variant,  
   STRING drug)  
2:   idList = []  
3:   results = []  
4:   genes = []  
5:   variants = []  
6:   drugs = []  
7:   if database is empty then  
8:     return []  
9:   else  
10:    len ← number of fields queried (0, 1, 2, or 3)  
11:    idList.push(IDs matching the query)  
12:    if len is 0 then  
13:      idList.push(all IDs in database)  
14:    else  
15:      genes.push(geneMapping[gene])  
16:      variants.push(variantMapping[variant])  
17:      drugs.push(drugMapping[drug])  
18:      idList.push(intersection of IDs in genes,  
19:        variants, and drugs)  
20:      for i ≤ length(uniqueEntries in contract storage)  
21:        do  
22:          for j ≤ length(idList) do  
23:            if idList[j] is ID of ith uniqueEntry  
24:              then  
25:                results.push(database[j])  
26:                convert results counts to percentages  
27:                j++  
28:                i++  
29:      return results
```



Algorithms

Algorithm 3 fastQuery Solution - Insertion

```
1: procedure INSERT(STRING gene, UINT variant,  
   STRING drug, STRING outcome, BOOL relation, BOOL  
   sideEffect)  
2:   entryIdentity.push(gene, variant, drug)  
3:   entryData.push(outcome, relation, sideEffect)  
4:   ID ← idKeeper[gene][variant][drug]  
5:   if database[ID] exists then  
6:     idKeeper[gene][variant][drug] ← counter  
7:     ID ← counter  
8:     database[ID].push(entryIdentity)  
9:     genes[gene] ← counter  
10:    variants[variant] ← counter  
11:    drugs[drug] ← counter  
12:    counter++  
13:    update database[ID] counts based on entryData
```

Algorithm 4 fastQuery Solution - Query

```
1: procedure QUERY(STRING gene, STRING variant,  
   STRING drug)  
2:   idList = []  
3:   results = []  
4:   genes = []  
5:   variants = []  
6:   drugs = []  
7:   if database is empty then  
8:     return []  
9:   else  
10:    len ← number of fields queried (0, 1, 2, or 3)  
11:    idList.push(IDs matching the query)  
12:    if len is 0 then  
13:      idList.push(all IDs in database)  
14:    else  
15:      genes.push(geneMapping[gene])  
16:      variants.push(variantMapping[variant])  
17:      drugs.push(drugMapping[drug])  
18:      idList.push(intersection of IDs in genes,  
19:        variants, and drugs)  
20:      for  $i \leq \text{length}(\text{idList})$  do  
21:        results.push(database[i])  
22:        convert results counts to percentages  
23:        i++  
24:      return results
```



- Project Design and Implementation

In detail, implementation is explained in IBC project reports.

Objective



- **To develop a robust method for storing, sharing and updating human aging genomics data among different entities like pharm companies, academics, researchers and healthcare professionals so that the data can be selectively shared while maintaining privacy.**
- Users own the data and can decide to share their genomics data anonymously, researchers get credit for their contributions, pharmaceutical companies get paid for their research and development.

March

- Ideation
- Fabric Fundamentals
- Project Framework
- CouchDB database creation

April

- Creation of Fabric Network
- Development of Smart Contract
- Fabric Security
- Deployment of Fabric Network

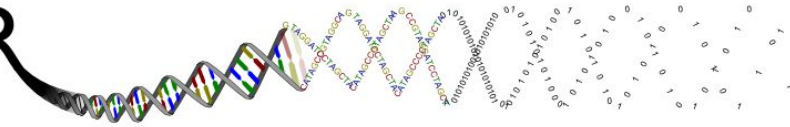
For plan described in detail please visit below reference link.

Ref: <https://github.com/Ghanendra19213/IBC/blob/main/IBC%20Project%20Plan.pdf>

HAGR

Human Ageing
Genomic Resources

at [senescence.info](https://genomics.senescence.info)



Databases

- GenAge Database of Ageing-Related Genes
- LongevityMap: Genetic association studies of longevity
- GenDR Database of Dietary Restriction-Related Genes
- DrugAge Database of Anti-Ageing Drugs

Why Aging data ?

In future nearly after 15 -20 years, research work in the field of ageing and longevity will mature enough to understand and help us to fight diseases and live healthier and longer lives as we age.

GenAge: The Ageing Genes



Benchmark database of genes related to ageing.

- GenAge is divided into genes related to longevity and/or ageing in [model organisms](#) (yeast, worms, flies, mice, etc.) and ageing related human genes.
- The section on [human ageing-related genes](#) includes the few genes directly related to ageing in humans plus the best candidate genes obtained from [model organisms](#). [Human genes](#) are thus considerably better annotated and include more information.



[Mouse](#)

[Download mouse genes](#)
[Download homologs from other model organisms](#)



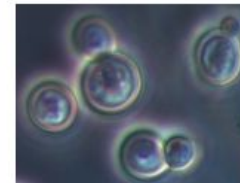
[Fruit fly](#)

[Download fruit fly genes](#)
[Download homologs from other model organisms](#)



[Roundworm](#)

[Download roundworm genes](#)
[Download homologs from other model organisms](#)



[Baker's yeast](#)

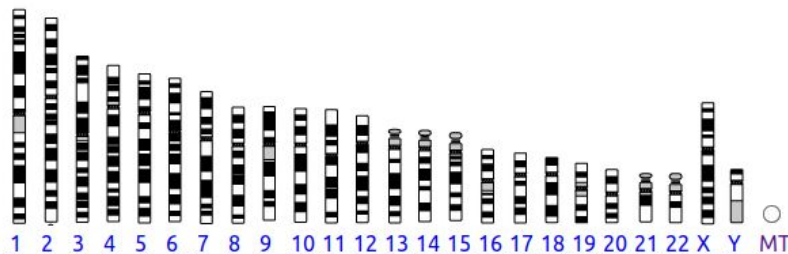
[Download baker's yeast genes](#)
[Download homologs from other model organisms](#)

LongevityMap: Human Longevity Genes

A database of human genetic variants associated with **longevity**.

- Negative results are also included in the LongevityMap to provide visitors with as much information as possible regarding each gene and variant previously studied in context of longevity.
- LongevityMap serves as a repository of genetic association studies of longevity and reflects our current knowledge of the genetics of human longevity.

Entries in chromosomes:



Longevity is the length of the life span independent of the biological aging process.

- Dietary restriction (DR), limiting nutrient intake from diet without causing malnutrition, retards age-related degeneration and extends lifespan in multiple organisms.
- DR induces multiple changes, yet its underlying mechanisms remain poorly understood. To facilitate research on the genetic and molecular mechanisms of DR-induced life-extension, we developed GenDR, a database of genes associated with DR.

DrugAge: Anti-Ageing Drugs



- DrugAge database contains an extensive compilation of drugs, compounds and supplements (including natural products and nutraceuticals) with anti-ageing properties that extend longevity in model organisms.
- Our focus is on drugs/compounds potentially impacting on ageing, and therefore drugs/compounds extending lifespan in disease-prone animals (e.g., cancer models) are excluded.

HyperLedger Fabric



It is an enterprise-grade permissioned distributed ledger framework.

- Depending on the problem statement and type of interaction between parties involved in business, we define few key features.

Participants, Assets and Transactions.

Participants (Organisations like Pharma company, Healthcare).

Assets (Genomic data) which gets transferred between entities.

Transactions (Reading/ Writing, Querying genomic data) which change the state of the ledger.

Why Hyperledger Fabric ?

In the case of genomics data storage, it provides off the shelf selective private data access between participants using CouchDB to store rich data queries.

Docker and docker-compose



```
ghanendra@ghanendra: ~$ HLF_VERSION=1.4.6
ghanendra@ghanendra:~$ docker pull hyperledger/fabric-peer:${HLF_VERSION} \
> && docker pull hyperledger/fabric-orderer:${HLF_VERSION} \
> && docker pull hyperledger/fabric-ca:${HLF_VERSION} \
> && docker pull hyperledger/fabric-ccenv:${HLF_VERSION}
1.4.6: Pulling from hyperledger/fabric-peer
Digest: sha256:d0a5c81f8f66276c542f6f642f3b144791e3a5a47cc919b0a17c335541c
Status: Image is up to date for hyperledger/fabric-peer:1.4.6
docker.io/hyperledger/fabric-peer:1.4.6
1.4.6: Pulling from hyperledger/fabric-orderer
Digest: sha256:f3071807c71ecdadc145c2fe8ae09f9bafcf8
Status: Image is up to date for hyperledger/fabric-orderer:1.4.6
docker.io/hyperledger/fabric-orderer:1.4.6
1.4.6: Pulling from hyperledger/fabric-ca
Digest: sha256:c7eed780d2908155ff2754b65aa3e4e3438f9
Status: Image is up to date for hyperledger/fabric-ca:1.4.6
docker.io/hyperledger/fabric-ca:1.4.6
1.4.6: Pulling from hyperledger/fabric-ccenv
Digest: sha256:697faea90b391c6a8b1b3aebb4fdeaf57aed1
Status: Image is up to date for hyperledger/fabric-ccenv:1.4.6
docker.io/hyperledger/fabric-ccenv:1.4.6
ghanendra@ghanendra:~$
```

```
ghanendra@ghanendra:~/fabric-sdk-py$ docker-compose
ose-2orgs-4peers-tls.yaml up
Building with native build. Learn about native build
cs.docker.com/go/compose-native-build/
Creating network "fixtures_default" with the default
Creating couchdb1_p1_org1 ... done
Creating couchdb0_p0_org1 ... done
Creating couchdb3_p1_org2 ... done
Creating orderer.example.com ... done
Creating couchdb2_p0_org2 ... done
Creating peer1.org2.example.com ... done
Creating peer0.org2.example.com ... done
Creating peer1.org1.example.com ... done
Creating peer0.org1.example.com ... done
```

```
ghanendra@ghanendra:~$ docker ps
```

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
1ad763325634	hyperledger/fabric-peer:latest		"peer node start"	11 days ago	Up 2 minutes	7050/tcp,
, 0.0.0.0:10053->7053/tcp	peer1.org2.example.com					
433a2b875d95	hyperledger/fabric-peer:latest		"peer node start"	11 days ago	Up 2 minutes	7050/tcp,
0.0.0.0:9053->7053/tcp	peer0.org2.example.com					
66e7f389f2e8	hyperledger/fabric-peer:latest		"peer node start"	11 days ago	Up 2 minutes	7050/tcp,
71c51113f10d	hyperledger/fabric-peer:latest		"peer node start"	11 days ago	Up 2 minutes	7050/tcp,
0.0.0.0:8053->7053/tcp	peer0.org1.example.com					
c58e3c7ae7ec	couchdb:2.3	couchdb2_p0_org2	"tini -- /docker-ent..."	11 days ago	Up 2 minutes	4369/tcp,
c00b410344e6	couchdb:2.3	couchdb1_p1_org1	"tini -- /docker-ent..."	11 days ago	Up 2 minutes	4369/tcp,
baa1bacb8f40	hyperledger/fabric-orderer:latest		"orderer start"	11 days ago	Up 2 minutes	0.0.0.0:7
36aa6fd5cd1b	couchdb:2.3	orderer.example.com	"tini -- /docker-ent..."	11 days ago	Up 2 minutes	4369/tcp,
f6c12460673d	couchdb:2.3	couchdb3_p1_org2	"tini -- /docker-ent..."	11 days ago	Up 2 minutes	4369/tcp,
		couchdb0_p0_org1				

```
ghanendra@ghanendra:~$
```

CouchDB Database



Used to store ageing genomic data

The screenshot displays the Apache CouchDB web interface. On the left, a sidebar contains navigation icons. The main area is divided into two sections. The top section, titled 'Databases', shows a table with two databases: 'longevity_data' (52.0 KB) and 'sample_genomic_data' (476 bytes). The bottom section, titled 'healthcarechannel...', shows a document view with a table of genomic data. The document view includes a sidebar with navigation options like 'All Documents', 'Run A Query with Mango', 'Permissions', 'Changes', and 'Design Documents'. The main content area shows a table with columns 'id' and 'gene~name'. The table contains six rows of data, each with a checkbox and a document icon.

Name	Size	# of Documents
longevity_data	52.0 KB	1
sample_genomic_data	476 bytes	1

id	gene~name
<input type="checkbox"/>	gene~nameAFZEJohn
<input type="checkbox"/>	gene~nameAPO1Dennis
<input type="checkbox"/>	gene~nameCDC13Munawar
<input type="checkbox"/>	gene~nameMEC1Donghoon
<input type="checkbox"/>	gene~nameRIF1Ghanendra
<input type="checkbox"/>	gene~nameTEL1Shubham
<input type="checkbox"/>	gene~nameTELOTanmay

Smart Contract

```
package main

import (
    "bytes"
    "encoding/json"
    "fmt"
    "strings"

    "github.com/hyperledger/fabric/core/chaincode/shim"
    pb "github.com/hyperledger/fabric/protos/peer"
)

// SimpleChaincode example simple Chaincode implementation
type SimpleChaincode struct {
}

type gene struct {
    ObjectType string `json:"docType"` //docType is used
    Id          int    `json:"id"`      //the fieldtags are
    Name        string `json:"name"`
    Population  string `json:"population"`
    Gene        string `json:"gene"`
    Size        int    `json:"size"`
}

type genePrivateDetails struct {
    ObjectType string `json:"docType"` //docType is used
    Name        string `json:"name"`    //the fieldtags are
    Age         int    `json:"age"`
    Variant     string `json:"variant"`
    Price       int    `json:"price"`
}
```

Ref: [smartContract_genomic_cc.go](https://github.com/hyperledger/fabric-samples/tree/main/smartcontract_genomic_cc_go)

```
// Invoke - Our entry point for Invocations
// =====

func (t *SimpleChaincode) Invoke(stub shim.ChaincodeStubInterface) pb.Response {
    function, args := stub.GetFunctionAndParameters()
    fmt.Println("invoke is running " + function)

    // Handle different functions
    switch function {
    case "initGene":
        //create a new gene
        return t.initGene(stub, args)
    case "readGene":
        //read a gene data
        return t.readGene(stub, args)
    case "readGenePrivateDetails":
        //read a gene private details
        return t.readGenePrivateDetails(stub, args)
    case "transferGene":
        //change owner of a specific gene information
        return t.transferGene(stub, args)
    case "delete":
        //delete a gene
        return t.delete(stub, args)
    case "queryLongevityMapByGene":
        //find genes from LongevityMap for user X using rich query
        return t.queryLongevityMapByGene(stub, args)
    case "queryAgeingDrugs":
        //find Ageing Drugs based on Compound name using rich query
        return t.queryAgeingDrugs(stub, args)
    case "getGenesByRange":
        //get genes based on range query
        return t.getGenesByRange(stub, args)
    default:
        //error
        fmt.Println("invoke did not find func: " + function)
        return shim.Error("Received unknown function invocation")
    }
}
```



Fabric Security



Used to store users enrolment credentials and certificates.

Use couchdbwalletStore to store credentials

```
In [7]: 1 # start first ca service before storing enrolment data.  
2 from hfc.fabric_ca.caservice import ca_service  
3 from hfc.fabric_network import couchdbwalletstore
```

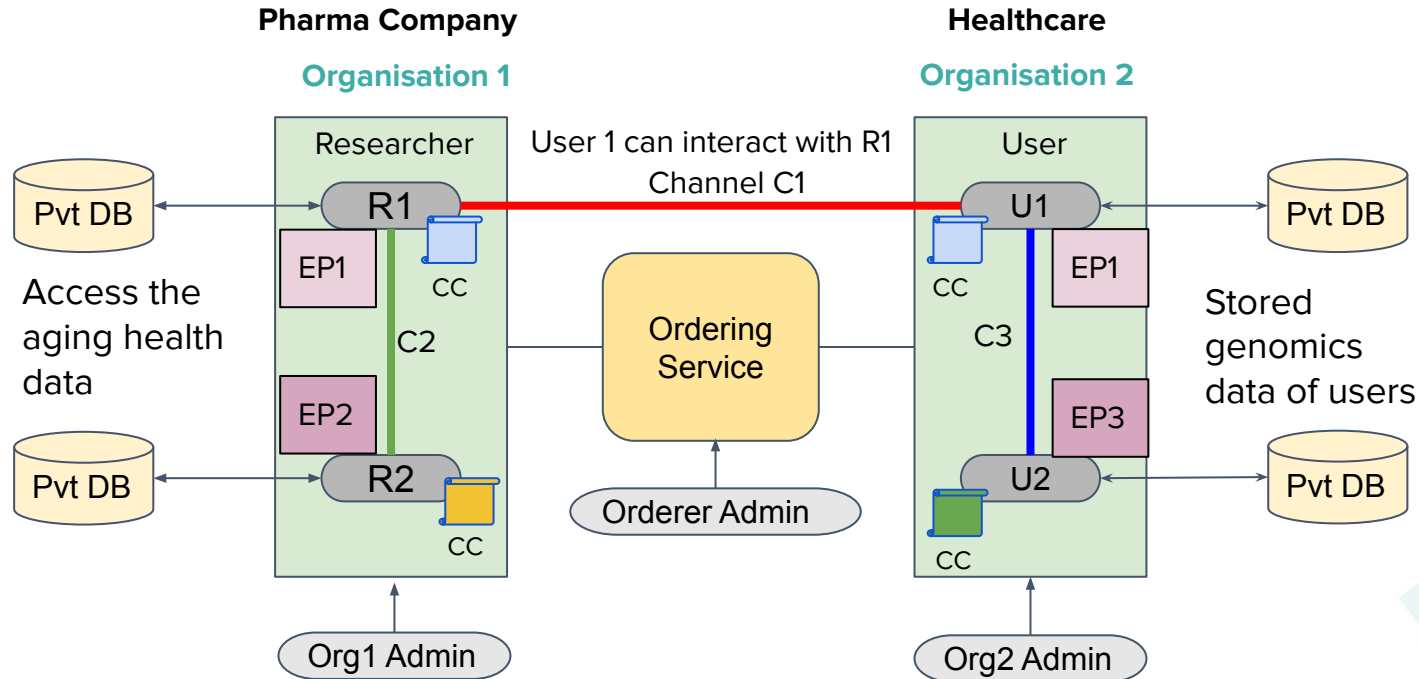
ghanendra@ghanendra: ~/fabric-sdk-py

```
Fabric-ca | 2021/03/27 20:45:01 [INFO] public key file location: /etc/hyperledger/fabric-ca-server/IssuerPublicKey  
Fabric-ca | 2021/03/27 20:45:01 [INFO] The Idemix issuer revocation public and secret key files already exist  
Fabric-ca | 2021/03/27 20:45:01 [INFO] private key file location: /etc/hyperledger/fabric-ca-server/msp/keystore/IssuerRevocationPrivateKey  
Fabric-ca | 2021/03/27 20:45:01 [INFO] public key file location: /etc/hyperledger/fabric-ca-server/IssuerRevocationPublicKey  
Fabric-ca | 2021/03/27 20:45:01 [INFO] Home directory for default CA: /etc/hyperledger/fabric-ca-server  
In Fabric-ca | 2021/03/27 20:45:01 [INFO] Operation Server Listening on [::]:40581  
Fabric-ca | 2021/03/27 20:45:01 [INFO] Listening on http://0.0.0.0:7054  
Out Fabric-ca | 2021/03/27 20:45:26 [INFO] signed certificate with serial number 197815981832100971458519712554531420015842158824  
Fabric-ca | 2021/03/27 20:45:26 [INFO] 172.17.0.1:59624 POST /api/v1/enroll 201 0 "OK"  
In Fabric-ca | 2021/03/27 20:45:26 [INFO] 172.17.0.1:59628 POST /api/v1/register 500 0 "Registration of 'user2' failed: Identity 'user2' is already registered"  
Fabric-ca | 2021/03/27 20:45:37 [INFO] signed certificate with serial number 536037026501892266510563587510044423870602046952  
Out Fabric-ca | 2021/03/27 20:45:38 [INFO] 172.17.0.1:59652 POST /api/v1/enroll 201 0 "OK"  
Fabric-ca | 2021/03/27 20:45:38 [INFO] 172.17.0.1:59656 POST /api/v1/register 500 0 "Registration of 'user3' failed: Identity 'user3' is already registered"  
Fabric-ca | 2021/03/27 20:45:50 [INFO] signed certificate with serial number 2858451152348967621494197115681053812602817552  
In Fabric-ca | 2021/03/27 20:45:51 [INFO] 172.17.0.1:59662 POST /api/v1/enroll 201 0 "OK"  
Out Fabric-ca | 2021/03/27 20:45:51 [INFO] 172.17.0.1:59666 POST /api/v1/register 201 0 "OK"  
Fabric-ca | 2021/03/27 20:45:51 [INFO] signed certificate with serial number 540175120214996901240207063803836597474968054102  
Fabric-ca | 2021/03/27 20:45:51 [INFO] 172.17.0.1:59670 POST /api/v1/enroll 201 0 "OK"  
In Fabric-ca | 2021/03/27 20:45:51 [INFO] signed certificate with serial number 173962058786617856077937163269748590989128420764  
Out Fabric-ca | 2021/03/27 20:45:51 [INFO] 172.17.0.1:59674 POST /api/v1/enroll 201 0 "OK"  
Fabric-ca | 2021/03/27 20:45:51 [INFO] 172.17.0.1:59678 POST /api/v1/revoke 200 0 "OK"
```

```
In [10]: 1 cdb_ws.exists('Molly')
```

```
Out[10]: True
```

Fabric Network Architecture



Invoke and Instantiate transactions

Perform read/write and query data.

- Project Demonstration

Demo of fabric network components - Docker containers, chaincodes, transactions and couchDB data storage over zoom or google meet video.

References



Project Link: <https://github.com/Ghanendra19213/IBC>

- Hyperledger Fabric [A Blockchain Platform for the Enterprise](#)
- SideDB pptx file [Privacy Enabled Ledger](#)
- Tacutu, R., *et al.* (2018) "Human Ageing Genomic Resources: new and updated databases." *Nucleic Acids Research* **46**(D1):D1083-D1090
- Gürsoy, G., Brannon, C.M. & Gerstein, M. Using Ethereum blockchain to store and query pharmacogenomics data via smart contracts. *BMC Med Genomics* **13**, 74 (2020).
<https://doi.org/10.1186/s12920-020-00732-x>
- Blockchain in Healthcare Today [Blockchain in Healthcare Today](#).

Thanks

Block creation time $\sim 1s$.

Q?

