

Network Biology

Coding Assignment 2

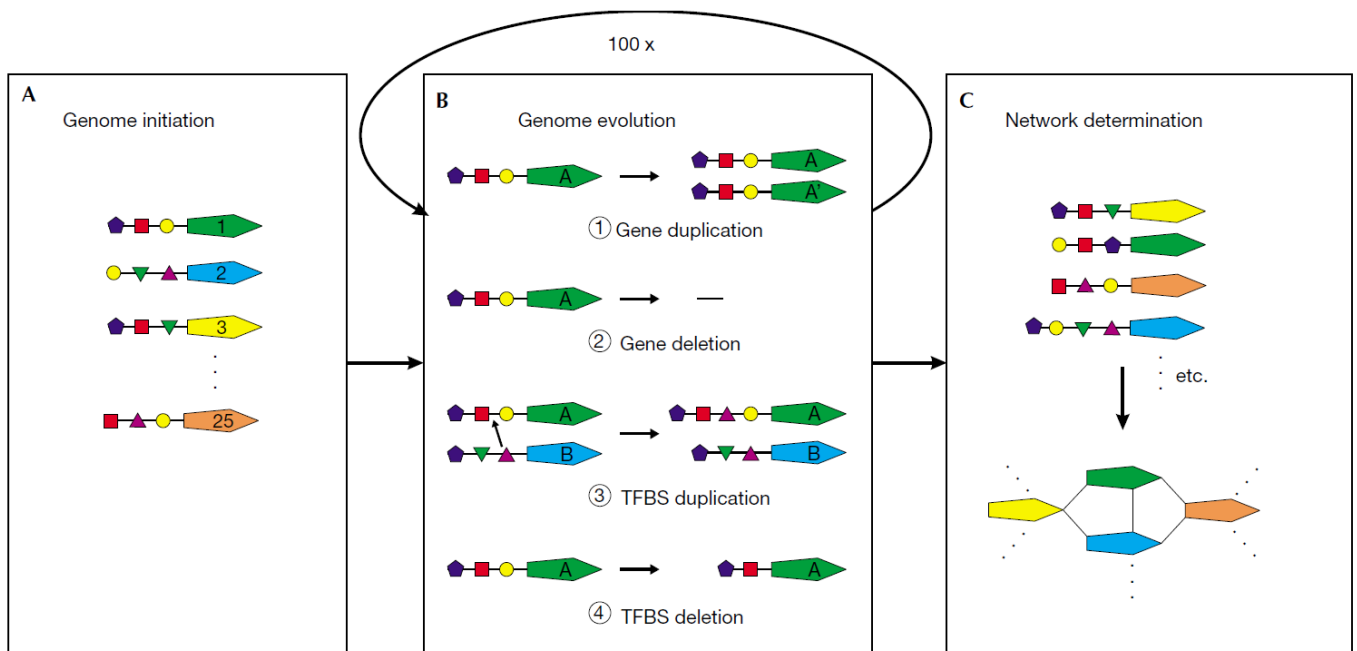
Note: You are responsible for the backup of data as well as results, which will be used for evaluation.

1. Implement the “evolutionary model of transcription regulation”.

- Start with an initial pool of small number of genes (say, 25) and transcription factor binding sites (say, 10) such that each gene has a fixed number of TFBS (say, 3).
- Implement genome evolution involving gene duplication, gene deletion, TFBS duplication, and TFBS deletion. Play with the probability values for each of these events.
- After a certain number of steps of evolution (say, 100), construct the co-expression network with the assumption that ‘two genes co-express if they share one or more TFBS’.

For specific initial values of number of genes and TFBS, probabilities used in the evolution step, and total number of steps in evolution: **[10+10+5]**

- Report the **final co-expression network** (image exported from Cytoscape).
- Plot the **degree distribution** of the networks for following cases: Gene Duplication \ll TFBS Duplication, TFBS Duplication \gg Gene Duplication, and Gene Duplication \approx TFBS Duplication.
- Provide your **conclusions** based on these results.



(‘The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model’, V van Noort, B Snel & M Huynen, *EMBO Rep.* 2004; 5(3): 274–279.)

2. Create Residue Interaction Graph (RIG) and Long-Range Interaction Network (LIN) models for **any five of the following thirty single-domain two-state folding proteins**, using the strategy listed below: 1hrc, 1imq, 1ycc, 2abd, 2pdd, 1aps, 1cis, 1coa, 1fkb, 1hdn, 1pba, 1ubq, 1urn, 1vik, 2hqi, 2ptl, 2vik, 1aey, 1csp, 1mjc, 1nyf, 1pks, 1shf, 1shg, 1srl, 1ten, 1tit, 1wit, 2ait, 3mef. [10]
- Download the PDB file.
 - Extract the atomic coordinates information (in any programming language, for extracting the coordinates flawlessly).
 - Further, extract the coordinates of $C\alpha$ atoms of each amino acid.
 - Using the data of representative $C\alpha$ atoms, **write a code** to create the RIG model using a cut-off of 7.
 - **Write a code** to compute the LIN model, using the threshold of 12 amino acids.
- a. Compute the characteristic path length (L) and clustering coefficient (C) of both RIG and LIN models for each of the above proteins. **Provide your results in a tabular form.**
- b. Write your observations about their topological properties and expected rate of folding.
3. Implement Bartoli's model of protein structure for the five proteins that you chose in response to Question 2. Compute the characteristic path length (L) and clustering coefficient (C) for the RIG model equivalent of Bartoli's models (for 100 instances).

Following is the procedure for generating contact maps using Bartoli's model:

- (i) Assign 1s to the first two diagonals (up and down the main diagonal) of the adjacency matrix in order to define the backbone contacts.
- (ii) Randomly select a pair of residues i and j with a probability that decreases linearly with the distance separating these residues in the protein sequence.
- (iii) Assign 1s to the entries of the adjacency matrix corresponding to all nine residue pairs generated by the Cartesian product of $\{i - 1, i, i + 1\} \times \{j - 1, j, j + 1\}$.
- (iv) Iterate the last procedure until the number of links in the random graph is *close to* those of the real protein.

- a. **Compare the results** of the RIG and LIN of the original structure to that of their Bartoli counterparts, **and write your observations**. Can you modify step (ii) to create contact maps that are closer to those from real protein structures? Report your results. [10]
- b. Compute the plot depicting the 'number of amino acid contacts made' and 'Cartesian distance between them' for all five proteins. Could it be possible to create a model to generate a contact map of a protein, that accounts for this feature? [5]

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer. This cheat sheet features the most important and commonly used Git commands for easy reference.

INSTALLATION & GUIs

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows

<https://windows.github.com>

GitHub for Mac

<https://mac.github.com>

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms

<http://git-scm.com>

SETUP

Configuring user information used across all local repositories

```
git config --global user.name "[firstname lastname]"
```

set a name that is identifiable for credit when review version history

```
git config --global user.email "[valid-email]"
```

set an email address that will be associated with each history marker

```
git config --global color.ui auto
```

set automatic command line coloring for Git for easy reviewing

SETUP & INIT

Configuring user information, initializing and cloning repositories

```
git init
```

initialize an existing directory as a Git repository

```
git clone [url]
```

retrieve an entire repository from a hosted location via URL

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

```
git status
```

show modified files in working directory, staged for your next commit

```
git add [file]
```

add a file as it looks now to your next commit (stage)

```
git reset [file]
```

unstage a file while retaining the changes in working directory

```
git diff
```

diff of what is changed but not staged

```
git diff --staged
```

diff of what is staged but not yet committed

```
git commit -m "[descriptive message]"
```

commit your staged content as a new commit snapshot

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

```
git branch
```

list your branches. a * will appear next to the currently active branch

```
git branch [branch-name]
```

create a new branch at the current commit

```
git checkout
```

switch to another branch and check it out into your working directory

```
git merge [branch]
```

merge the specified branch's history into the current one

```
git log
```

show all commits in the current branch's history



INSPECT & COMPARE

Examining logs, diffs and object information

git log

show the commit history for the currently active branch

git log branchB..branchA

show the commits on branchA that are not on branchB

git log --follow [file]

show the commits that changed file, even across renames

git diff branchB..branchA

show the diff of what is in branchA that is not in branchB

git show [SHA]

show any object in Git in human-readable format

TRACKING PATH CHANGES

Versioning file removes and path changes

git rm [file]

delete the file from project and stage the removal for commit

git mv [existing-path] [new-path]

change an existing file path and stage the move

git log --stat -M

show all commit logs with indication of any paths that moved

IGNORING PATTERNS

Preventing unintentional staging or committing of files

```
logs/  
*.notes  
pattern*/
```

Save a file with desired patterns as .gitignore with either direct string matches or wildcard globs.

git config --global core.excludesfile [file]

system wide ignore pattern for all local repositories

SHARE & UPDATE

Retrieving updates from another repository and updating local repos

git remote add [alias] [url]

add a git URL as an alias

git fetch [alias]

fetch down all the branches from that Git remote

git merge [alias]/[branch]

merge a remote branch into your current branch to bring it up to date

git push [alias] [branch]

Transmit local branch commits to the remote repository branch

git pull

fetch and merge any commits from the tracking remote branch

REWRITE HISTORY

Rewriting branches, updating commits and clearing history

git rebase [branch]

apply any commits of current branch ahead of specified one

git reset --hard [commit]

clear staging area, rewrite working tree from specified commit

TEMPORARY COMMITS

Temporarily store modified, tracked files in order to change branches

git stash

Save modified and staged changes

git stash list

list stack-order of stashed file changes

git stash pop

write working from top of stash stack

git stash drop

discard the changes from top of stash stack

GitHub Education

Teach and learn better, together. GitHub is free for students and teachers. Discounts available for other educational uses.

✉ education@github.com
🌐 education.github.com

EDUCATION

Master of Business Administration (Specialization in Foreign Trade), Banaras Hindu University
Bachelor of Science (PCM), Mahatma Gandhi Kashi Vidyapith University

Aug 2019 — Jun 2021
Jun 2015 — Jun 2018

SKILLS

- Natural Problem-solving abilities
- Strong presentation skills and speaking ability; comfortable with interacting with senior leadership
- Ability to prioritize and identify critical activities
- Proficiency in MS Excel and MS Word
- Possessing a proven track record of accuracy and strong attention to detail
- Pro-active and willing to work in a team or autonomously
- Capable in delivering consistently high standards of written and verbal communication
- Strong analytical skills with the ability to look objectively and bring disciplined analysis to date and situations to ensure that all outcomes are thoroughly assessed

EXPERIENCE

ESG Research Analyst (FTE)

Oct 2021 — Present

SustainoMetric

Varanasi

- Data mining, conduct secondary research, identify, aggregate, and analyze information for ESG indicators using reports like sustainability reports, annual reports, Form10-K, corporate governance documents, Proxy Statement, Bylaws, Articles of Association, Code of Conduct, Suppliers Code of Conduct, Policies, etc.
- Sustainability research and analysis of companies across multiple sectors on ESG indicators using comprehensive desk-based research.
- Insert relevant corporate information in the database. Extract information on economic activity quantification from the company reports and insert the respective information in the database.
- Track, summarize, and assess global news for the ESG controversies about companies.
- Worked on projects like FactSet, OWL ESG, OWL Controversy, Carbon data collection, and Ethos.
- I handled a team of 3-4 people, where I trained them and audited their work.
- Gained a deep knowledge of ESG Frameworks and Standards like CDP, GRESB, DJSI, SASB, CDSB, IIRC, TCFD, GRI, etc.

Business Analyst (Intern)

May 2020 — Aug 2020

UniConverge Technologies Private Limited

Noida

- My work was to make strategies to generate leads for the internship cum training program

DIPLOMA AND CERTIFICATION

Banaras Hindu University

Aug 2020 — Aug 2021

- Diploma in Corporate Governance

Corporate Finance Institute

- Introduction to ESG

NSE India

- Financial Statement Analysis

UNCC: e-Learn

- Introduction to Sustainable Finance

NPTEL

- Financial Accounting

IGNOU

Jan 2023 — Jan 2024

- Diploma in Environmental Management and Law

Nithin Reddy, Pod Lead, Growth and Engagement, Navi Technologies, Bangalore, India
8128244670, nithinreddyt@gmail.com

TECHNOLOGIES AND SKILLS

Java, Spring boot, GoLang, System Design, MYSQL, Elasticsearch, Couchbase, NodeJS, AWS services, MongoDB, Kafka, Microservices, Distributed Systems.

As an experienced professional with a proven track record of leading teams and managing complex projects, I bring a unique blend of technical expertise and leadership skills to any organization. With extensive experience in design, implementation, and cross-team dependency management, I have a deep understanding of what it takes to ensure successful project delivery and achieve business goals.

WORK EXPERIENCE

Navi, Pod Lead - Growth and Engagement Team (Sept 2021 - Present)

- Responsible for design, code reviews, technical decisions, and uptime of multiple services our team owns.
- Being the technical face of the team I work closely with the Engineering head, EMs, Product managers, and other key stakeholders at Navi.
- As a Pod Lead, I **mentor and guide 9 other people** on the team.
- Actively part of **hiring** SDE-3, SDE-2, and SDE-1.
- Solely responsible for Navi Apps home page **uptime and latency optimizations**.
- **Feeds Service:** Was solely responsible for the **Design, Project Planning, Dependency Management, Implementation**, and release planning of Feeds Service.
 - Feeds - responsible for all key real estate of Navi App, Home screen, Cross-sell, and Upsell screens.
 - Service uses a config-based approach for screens and serves personalized cards(UI widgets) to the users based on their profile and current journey.
 - Resiliency, Latency, and Scale were key factors for this service as it serves Navi's home page. Improved home page latency from 750ms to 450ms.
 - Cards can be modified, added, and deleted on the go without any app or backend release.
 - Provided capabilities for Product managers to create, update and experiment on cards, thereby reducing dependency on Developers.
 - Responsible for code reviews, mentoring, and guiding 3 other developers in the team.

- Designed and implemented the **Navi Referral Engine** for user acquisition. Helped in reducing CAC by 60%.
- **Customer Data Platform:** Responsible for Design, Execution, and Road map for the Navi CDP platform. This system is being built to support Navi's Marketing use cases and segmentation use cases.
 - This System involves streaming architecture where we use Apache Flink, and then Apache Pinot as our data store.
 - Responsible for gathering requirements from the marketing team and planning and prioritizing the features.
 - Responsible for mentoring and guiding 2 other developers in the project.
- **Rewards, Digital Gold, Stories:** Here I'm responsible for system design, and code reviews. I mentor engineers who actively work on implementation.
- **Financial Profile Service:** Built a service to track the financial profile of Navi customers.
 - I was responsible for interacting with third-party financial data providers, and account aggregators. I help PMs to technically evaluate the right vendor for a given financial data.
 - Owned and partially implemented this project and mentored and guided 4 other developers involved in this project.
- Being a senior member of the team, I actively help and am involved in all app uptime issues across teams at Navi.

Mindtickle, Senior Software Developer - Distributed Systems (Sept 2018 - Sept 2021)

I worked for a distributed database team, where I was responsible for multiple services, databases (MySQL, Elasticsearch), and data pipelines. Also driven storage-related projects and promoted best practices across teams. Hiring, onboarding, Code reviews, and mentoring were a few other responsibilities.

1) **TickleDB:**

- Implemented a distributed data store, in which I was responsible for Data Plane, Control Plane, Routing, Auto Failover, Point-in-time Recovery, and Backup.
- Used Golang for developing microservices and MySQL for storing data.
- Solved challenging problems like - Multi-Tenancy, Horizontal scaling, Connection management, Database Abstraction, Multi-tenancy, Query Monitoring, Connection Monitoring, and Alerting.
- Achieved Auto failover of MySQL nodes - the system can now recover in 2-3 seconds.

- e. As the primary owner of TickleDb, I'm responsible for gathering requirements and helping decide on a road map.
- f. Scaled **TickleDB** from 0 to 50M req/day and it serves as a storage solution for 40 odd microservices across 8 teams.

2) **Elasticsearch:**

- a. I Managed a 30-node cluster with 900 indices. Responsible for cluster performance and health, tuning JVM heap, and scaling.
- b. Solved performance-related problems to support large numbers of indices at scale.
- c. Also identified concurrent index refresh problems, Memory constraints of shards, and Thread Contention issues while concurrent writes, by diving into the Elasticsearch 5.4 codebase
- d. Scaled ES cluster from 80M to 400M req/day.

3) **MT-Kafka-Connect:**

- a. I was responsible for building data pipelines on **Couchbase** and **MySQL**.
- b. Helped teams across Mindtickle to consume CDC events. System also includes replaying events in case of a disaster.

Lyearn Inc - *Lead Backend Developer, Dec 2016 - Sept 2018*

- a. Developed backend services from scratch, I was the first backend developer at Lyearn.
- b. Implemented Message driven Architecture using SQS - scaled up course enrollments.
- c. I implemented the Reporting and Admin services of the platform in a scalable manner, with full-text search capabilities, and various sort and filter capabilities using Elasticsearch as the data source.
- d. Established coding standards and code review process. Onboarded new developers and was also leading System Design, Planning, and Project management

Lyearn Inc - *Software Developer Intern June 2015 - Nov 2016*

Owned MongoDB and Elasticsearch. Developed backend services. Implemented user and user group hierarchies using Neo4j. Implemented GraphQL layer.

EDUCATION

Dhirubhai Ambani Institute of Information and Communication Technology - *B.Tech*, 2013 - 2017, CGPA: 8.51