AVPPIN Software Architecture

**1. Context**

The **A**nalysis and **V**isualization of **P**rotein-**P**rotein **I**nteraction **N**etworks (**AVPPIN**) tool allows users to store, compare, and visualize protein-protein interaction (PPI) networks on the web. This tool will enable population genetics of PPI networks between affected and unaffected individuals across multiple data sets to find new insights for disease. Users can access AVPPIN from a web browser to upload PPI data, perform statistical analysis on one or more PPI networks, create plots, and download results.

The purpose of this document is to:

* Define the key functions, organization, software components, database, and security model for AVPPIN
* Satisfy requirements for BMSE Assignment 2
* Provide a development guide for a similar system at the Bin Zhang Lab @ ISMMS

**2. Functional overview**

1. What does the system do?

The AVPPIN will provide many functional features to enable the efficient analysis of PPI networks:

* Ingestion of PPI networks from a variety of formats into a common standard
  + Extracting PPI network files from a variety of formats (.txt, .gexf, .cys)
  + Detect improperly formatted files, including unknown fields and proteins
  + Transforming PPI networks into a standard vocabulary (e.g. Ensembl ID) to facilitate comparison from a large set of alternate vocabularies (e.g. gene name)
* Storage of PPI network data in a relational database (e.g. MySQL) to create a data warehouse for future research
  + Loading PPI networks into the relational database indexed by an ID that facilitates future analysis
  + Includes both public (accessible to all users) and private data (accessible to a specific user(s))
* Annotating PPI networks with Gene Ontology (GO) information to understand PPI network functional significance
  + GO classifies groups of proteins as belonging to certain families by functional pathway or enrichment
* Comparison of PPI networks across datasets using statistical functions to find new insights about health and disease
  + These may be coded in at least either Python and R
  + “Does an affected network differ statistically from an unaffected network? Is a sub­network of proteins that differs between an affected network and an unaffected network over or under­represented in protein classification categories? Other analyses will be added as needed.”
* Storage of study and network metadata, indexed by one or more ID(s)

In addition, the AVPPIN will provide a web-browser based interface where users may access all of the mentioned functions as well as the following:

* Create user accounts, log in and out, upload and download data, load statistical functions, and perform, save, and view analyses
* Visualize gene networks in the web browser

1. Users / Stakeholders

The AVVPIN tool has three types of users:

1. Anonymous users: Anybody may view publically accessible content on the site but cannot interact with AVPPIN functions
2. Authenticated users: Authorized bioinformaticians who are at least affiliated with Mount Sinai may create user accounts to store, visualize, and compare PPI networks, and share analyzes among members of their group(s)
3. Administrators (super-users): Site administrators will be able to manage all users and groups, content, and functions provided by the AVPPIN to users

The AVPPIN tool will serve various stakeholders, including:

1. Bioinformaticians in research labs at Mount Sinai who assist research geneticists with their analyses
2. Research labs at Mount Sinai that will publish results from analyses
3. General public, including other research labs, that will view and download results

**3. Software architecture**

1. Overview

The AVPPIN is a collection of the following software components that interact with each other to provide the described functions:

1. Python backend controller: core functions
2. MySQL database: data storage
3. Asynchronous task engine: performing analysis and visualization
4. Web server (e.g. Apache2): web functions

Additional details about each component are provided later in this section.

1. Inputs/Outputs

The AVPPIN inputs four types of data:

1. PPI Networks: A PPI network represents each protein as a node labeled with its name, and a relationship between a pair of proteins as an edge between their nodes. The relationships are categorical. One kind of relationship is 'interact', which is represented as an undirected edge. Other relationships are ‘up­regulates’ and ‘down­regulates’, which are represented as directed edges from the regulatory protein to its target. Additionally, each edge in the network may have additional properties associated with it, including a ‘weight’ representing the strength of the interaction and a ‘p-value’ representing the probability of a non-null interaction.

The PPI networks contain research data that is not Protected Health Information (PHI) under HIPAA. Two types of networks are used:

* Reference PPI networks for particular human tissues obtained from populations of unaffected individuals, which are called “unaffected networks”.
* One or more PPI networks for the same tissue obtained from samples of subjects who have an illness -- affected individuals. These are called 'affected networks'.

1. Gene Ontologies: This information classifies proteins as members of certain genetic pathways or biological activities, e.g. as provided by the Gene Ontology system.
2. Gene Identities: This information matches proteins to one or more genes, transcripts, and genomic regions across different data sets and ID systems, e.g. as provided by Ensembl.
3. Metadata: The tool accepts metadata about each PPI network uploaded to the system as well as the study it belongs to. The tool also accepts metadata about users and their use of the AVPPIN (e.g. user credentials and the studies and networks that belong to them).
4. Organization of AVPPIN

The AVPPIN interacts with many other systems:

* MySQL instance on Minerva: Data generated by use of the AVPPIN and user profile data will be transformed to a relational database structure and deposited in a MySQL instance running on the Minerva supercomputing cluster
* Mount Sinai Hospital (MSH) Active Directory (AD): For the purposes of the AVPPIN, permitted users will be active MSH employees with logins
* Minerva LSF Queue: Long-running tasks/analyses will be sent to Minerva’s LSF queuing system for batch processing

The AVPPIN requires versioned data definitions from other systems:

* Ensembl (ensembl.org/): Provides information for the classification of proteins, genes, and transcripts in many releases that change over time
* MSigDB (software.broadinstitute.org/gsea/msigdb/): Provides an annotated Gene Ontology database in many releases that change over time

1. Components

The AVPPIN software flow will follow the State-Action-Model (SAM) design pattern: that is that users will perform functions through the view that are executed as actions to update the model and return the next state.

1. Components required
   1. What can be reused? Existing python packages
   2. What needs to be custom-built?
2. Database
   1. Tables
   2. Foreign keys relationships to support (remaining keys need not be provided)
3. Security
4. Block diagram of architectural components and interactions