Workflow for PPI Network Analysis and RWR Essential Protein Prediction (Detailed with Example and Formulas)

1. **User Input:**
2. User enters a disease name, e.g., `Type 2 Diabetes`.
3. User selects species, e.g., `Homo sapiens`.
4. This input triggers the app to fetch disease-related genes.
5. **Query Open Targets:**
6. GraphQL query is sent to Open Targets API.
7. Returns a list of genes associated with the disease and their association scores.
8. Example output: `[('TLR4', 0.8), ('TNF', 0.75), ('AGER', 0.72)]`.
9. **Map Genes to STRING IDs:**
10. Gene symbols are mapped to STRING database IDs needed for PPI data.
11. Example mapping: `{'TLR4': '9606.ENSP00000363089', 'TNF': '9606.ENSP00000398698', 'AGER': '9606.ENSP00000364210'}`
12. **Fetch PPI Data:**
13. For each STRING ID, the app fetches interaction partners from STRING.
14. Edges are filtered by confidence score (>= 400).
15. Example edge: `('TLR4', 'TNF', score=0.85)`.
16. **Build Network Graph:**
17. Nodes = proteins, Edges = interactions.
18. Compute network statistics: number of nodes, edges, density, average degree.
19. Example: `Nodes=89, Edges=678, Density=0.17`.
20. **Compute Centrality Measures:**
21. Degree centrality: number of connections per node.
22. Betweenness centrality: frequency a node lies on shortest paths.
23. Closeness centrality: inverse average distance to all other nodes.
24. Eigenvector centrality: importance based on connections to important nodes.
25. PageRank: weighted importance using network connectivity.
26. **Select Top Proteins by Centrality:**
27. For each metric, pick top 10 proteins.
28. Example: `Top Degree: ['TLR4', 'TNF', 'AGER', ...]`

29. **Weighted Centrality Score (Essentiality):**

30. Normalize centrality scores (0–1).
31. Weighted sum formula:

$$Essentiality = 0.30 * Degree_{norm} + 0.25 * Betweenness_{norm} + 0.20 * Closeness_{norm} + 0.15 * Eigenvect$$

32. Sort to find top 10 essential proteins.

33. **Random Walk with Restart (RWR):**

34. Convert network to adjacency matrix `A`.
35. Normalize columns: $M = A/extcolumnsums$.
36. Initialize seed vector p_0 with 1/N for each seed.
37. Iterate formula until convergence:

$$p^{(t+1)} = (1 - r)Mp^{(t)} + rp_0$$

    - `r` = restart probability (e.g., 0.7)
    - `p^{(t)}` = probability vector at iteration t
38. Proteins visited more frequently by the random walk get **higher RWR scores**.
39. Example top RWR proteins: `[TLR4:0.061, TNF:0.043, AGER:0.041]`.

40. **Display RWR Results:**

    - Show top 10 proteins with RWR scores and STRING descriptions.
    - Bar plot visualizes protein importance.

41. **Visualizations:**

    - Network plot colored by centrality category:
    - Red = Degree hubs, Orange = Betweenness spreaders, Green = Closeness connectors, Purple = Eigenvector influencers, Blue = others.
    - KDE plots for centrality distributions to understand network structure.

42. **Summary:**

    - Combine weighted centrality and RWR scores to prioritize proteins likely essential for the disease.
    - Output tables and plots allow users to explore proteins, their network position, and relevance to disease.

Example Workflow Flow: - Input: `Type 2 Diabetes`, `Homo sapiens`. - Query Open Targets → get `['TLR4', 'TNF', 'AGER', ...]`. - Map to STRING IDs → `['9606.ENSP00000363089', '9606.ENSP00000398698', '9606.ENSP00000364210', ...]`. - Fetch PPI → build network of 89 nodes, 678 edges. - Compute centralities → top degree hubs: `['TLR4', 'TNF', 'AGER', ...]`. - Weighted essentiality → top 10 proteins. - Run RWR → top 10 proteins by network proximity to seeds. - Display network plots, RWR scores, and protein descriptions.