NetworkSimDB: Simulations-Datenbank für (metabolischen) Netzwerke mit angeschlossener Visualisierung

Zusammenfassung

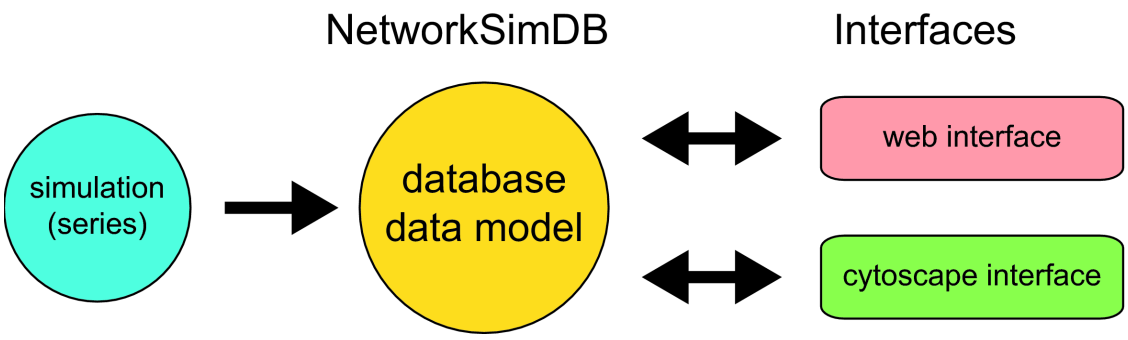
Entwicklung und Implementation eine **Datenbank zur Speicherung und Visualisierung von (Netzwerk-)Graphen mit assozierten Kanten- und Knotenattributen**.

Anwendungen sind die Speicherung und Auswertung von Simulationsergebnissen basierend auf Flussmethoden (wie FBA) oder kinetischen Methoden in metabolischen Netzwerken wie diese beispielsweise im Project Virtual Liver auftreten.

NetworkSimDB bildet ein DataWarehouse für Simulationen der Systembiologie, ist aber nicht auf diese beschränkt. Durch die Verwendung von **NetworkSimDB** kann die stets zwischen Simulation und Auswertung auftretende Lücke teilweise geschlossen werden. Simulationsergebnisse können automatisiert abgespeichert und leicht ausgewertet werden. Im Zusammenspiel mit nachgeschalteter Visualisierung (FluxViz in Cytoscape oder browserbasiert) wird so der Auswertungsprozess und das Verständnis der Simulationen enorm beschleunigt.

Weiterhin sind die Simulationsergebnisse dauerhaft und zentral mit Annotierten Informationen verfügbar. Hierdurch wird die Wiederverwertung bereits durchgeführter Simulationen enorm gesteigert.

@Michael Im Project Virtual Liver kann NetworkSimDB als Simulationlayer unabhängig von HepatoBase fungieren. Durch die Annotation beliebigen Informationen an Knoten und Kanten ist es äusserst einfach die Simulationslayer von NetworkSimDB mit der Ebene der Knowledgebase (HepatBase) anzureichern und den Netzwerkeigenschaften biologische Bedeutung zuzuweisen. Das selbe gilt selbstverändlich auch für andere exterene Datenbanken.



Datenmodell

Das Datenmodell muss in der Lage sein

- Graphen zu repräsentieren (Multigraph, directed, nondirected, ...)
- Knoten und Kanten im Graphen zu repräsentieren
- beliebige Knoten- und Kantenattribute aufzunehmen (diese müssen zuvor nicht bekannt sein, sondern können ad hoc für die jeweiligen Simulationen definiert werden → möglichst große Flexibiltät in der Anwendung, keine Einschränkung auf bestimmte Netzwerke)
- Simulation zu repräsentieren (mit Datum, Ziel Simulation, Simulations-Settings, ...)
- Simulations-Serien (oftmals batch Simulationen, die inhaltllich zu einer Serie zusammengehören, da nur sehr geringer Unterschied zwischen den Simulationsbedingungen → beispielsweise kinetische Zeitreichen, oder Variation eines Constraint in FBA) zu repräsentieren
- Multi-User fähig sein und zwischen privaten und public Daten unterscheiden können (d.h. Sessions,

Logins)

- Klassifizieriung der Simulationen basierend auf den Knotenattributen

Datenbank

- Postgres oder sehr leichtgewichtige Datenbanklösung

Cytoscape Interface

- aus Cytoscape heraus muss auf die Datenbank zugegriffen werden können und die gespeicherten Netzwerke samt Knoten- und Kantenattributen eingelesen werden können
- GUI für die Auswahl von Simulationen und Netzwerken (aus Cytoscape heraus zugriff auf die Datenbank oder Webinterface) - direkte Datenbankabfrage vs. Informationsabruf aus dem Webinterface

Webinterface

- Übersicht über die Inhalte in der Datenbank
- Suchfunktionen
- Fileupload (Simulationen) & Formulare für die Annotation der Simulationen

Tools & Sprachen

- Cytoscape Interface: Java (Cytoscape Java basiert)
- Datenbank: Postgres oder evt. leichtgewichtige Alternative, die auch direkt in Cytoscape integrieren werden kann
- Webinterface: Django vs. JavaEE (Vorteile, Nachteile)
  - Django sehr einfach und schnell implementiert, kein allzu großer Aufwand
  - Wo soll die Funktionalität implementiert werden → Graphenalgoritmen? Aus Cytoscape sollte man in der Lage sein auf die Webinterface-Funktionalität zuzugreifen. Daher wahrscheinlich eine vollständige Java Lösung zu bevorzugen
- lokale Datenbank vs. Webdatenbank

Methodik

- mir geht es v.a. auch darum in einem Team zusammenzuarbeiten und die Probleme zu sehen und zu lösen, die dabei auftreten können.
  - Definition von Interfaces für die Berührungsbereiche (Überlappungsbereiche der Aufgaben), wie sollen die Klassen / Module zusammenarbeiten
  - Content management system, versioning System → wie geht man damit um, best practises
  - Freiheit in den Entscheidungen vs. gemeinsame Linie für das Projekt finden
  - test driven development ??? Viele tests, die das gesamte System zusammenhalten und für Stabilität sorgen.
  - Projektplanung und Projekttreffen
  - gute Dokumentation

Provisorische Übersicht Aufgaben und Beteiligung

	Priority	Dependency	Matthias König matthias.koenig@charite.de	Torsten Houwaart toho@gmx.de	Michael Weidlich michael.weidlich@charite.de
Datenmodell	1		x	x	x
Datenbank Implementation	1	Datenmodell	x		x
Datenerzeugung (Beispiele FBA)	2		x		
Datenspeicherung	2	Datenbank Datenerzeugung	x		
Web Interface	3	Datenbank (Datenspeicherung)		x	?
Cytoscape Interface	3	Datenbank Datenspeicherung	x		
Cytoscape Visualisierung	3	Cytoscape Interface FluxViz	x		
Web Visualisierung	4	Web Interface	?	x	?

Weiteres Vorgehen

- alle hier vorgestellten Punkte sind provisorisch, nichts ist in Stein gemeiselt.
- Treffen aller Beteiligten und Diskussion des weiteren Vorgehens
- Ausarbeitung Arbeitsplan und Projektzeitplan
- Klären, was sich die einzelnen von dem Projekt versprechen, was jeder erwartet
- Zeitrahmen, wer kann wieviel Zeit investieren