NTNU
Det skapende universitet

Felles ordbok for identifisering av protein-/genforekomster og interaksjoner i biomedisinske artikkelsammendrag.

muligheter og utfordringer

Mari Lie Hæreid

Master i informatikk
Oppgaven levert: Mai 2007
Hovedveileder: Herindrasana Ramampiaro, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

Sammendrag

Formålet med denne oppgaven var å se på muligheten for samling av flere annoterings- og interaksjonsdatabaser, for dermed å kunne forbedre identifiseringen av protein-/genforekomster og -interaksjoner i biomedisinske artikkel-sammendrag.

For å underbygge vurderingene foretatt i oppgaven er en prototype implementert. Den viser hvordan man kan hente ut aktuell informasjon fra forskjellige annoterings- og interaksjonsdatabaser, og lagre dem i en felles relasjonsdatabase. I prototypen blir relasjonsdatabasen indeksert og det er implementert muligheter for tekstsøk mot indeksen. Resultatet av søk viser protein-/gennavn og/eller -synonymer fra relasjons-databasen, samt tilleggsinformasjon som symbol/id, interaksjoner med andre protein/gen og kryssreferanser mellom annoteringsdatabasene.

For å teste om prototypen fungere som ønsket og at relasjonsdatabasen lagrer informasjon på en tilfredsstillende måte, er det hentet ut testsett fra annoterings- og interaksjonsdatabasene.Testresultatene viser at informasjon blir lagret, indeksert og kan gjenfinnes på en måte som oppfyller de stilte kravene.

Vurderingene som er foretatt, sammen med prototypen, viser at en felles relasjonsdatabase vil være mulig og kan fungere bra for identifisering av protein-/genforekomster i artikkelsammendrag. Til tross for dette er det mange utfordringer som gjenstår før en slik samling vil fungere optimalt. Blant annet er mange av annoterings- og interaksjonsdatabasene forskjellige både i struktur og innhold, samt at det av forskjellige grunner kan være vanskelig å hente ut data fra dem. Dette er også mye av grunnen til at lite forskning er utført innenfor dette området og at det må vurderes hvorvidt det er hensiktsmessig og forsette arbeidet med å utvikle en slik samling, med tanke på ressursbruk og hva man kan få igjen for det.

Forord

Denne masteroppgaven har blitt skrevet som en avslutning på mastergraden i informatikk ved Norges Tekniske og Naturvitenskaplige Universitet. Arbeidet med oppgaven har foregått i perioden januar 2006 til juni 2007.

Jeg har gjennom denne oppgaven lært mye om informasjonsgjenfinning, både generelt og innen bioinformatikk. Jeg hadde lite kunnskap om området bioinformatikk da jeg startet opp og har hatt stor nytte av tidligere forskning innenfor området identifisering av biomedisinske termer i artikkelsammendag. Etter å ha skrevet denne oppgaven sitter jeg igjen med inntrykk av at dette er et stort og forholdsvis utoversiktlig område og jeg forstår hvorfor det finnes så mange forskjellige tidligere tilnærmelser. Jeg har også en forståelse for hvorfor det er foretatt forholdsvis lite forskning innen det området oppgaven min omfavner, men mener likevel det er et viktig område, det burde fokuseres mer på.

Jeg vil takke min veileder, førsteamanuensis Heri Ramampiaro, for god veiledning. Ønsker også takke stipendiat Yan Hua Chen og Knut Ola Bjerke for nyttige innspill.

Trondheim, 1.juni 2007

Mari Lie Hæreid

Innhold

Figurliste

Tabelliste

1 Introduksjon

1.1 Motivasjon

Det finnes mange annoteringsdatabaser med mye forskjellige informasjon om protein og gener. Dessuten finnes det egne databaser for protein og gen interaksjoner. Mange av disse databasene inneholder svært mye informasjon og kun noe av denne informasjonen er nyttig for identifisering av protein-/gennavn og -interaksjoner i biomedisinske artikkelsammendrag. I tillegg har mange av de forskjellige annoterings- og interaksjonsdatabasene forskjellig struktur og format, samt at det er stor forskjell på hvorvidt og hvordan data kan hentes ut fra dem. Av den grunn blir ofte kun en enkelt annoterings- eller interaksjonsdatabase brukt i hver tilnærmelse til informasjonsgjenfinning i artikkelsammendrag.

For å kunne forbedre identifiseringen av protein-/genforekomster og -interaksjoner i artikkelsammendrag vil det derfor være hensiktsmessig å samle innholdet i flere annoterings- og interaksjonsdatabaser. På den måten kan man oppnå en mer fullstendig samling, der informasjonen nyttig for gjenfinningsformålet er samlet.

1.2 Problemspesifikasjon

På bakgrunn av utfordringene nevnt i kapittel 1.1 vil jeg i denne oppgaven se på mulighetene for å samle informasjon fra allerede eksisterende protein-/gen-annoterings- og interaksjonsdatabaser og dermed lage en samling som egner seg spesielt godt for identifisering av protein-/gennavn og -interaksjoner i artikkel-sammendrag. Jeg tar for meg noen utvalgte annoterings- og interaksjonsdatabaser og vurderer hvorvidt og på hvilken måte de er egnet til å kunne brukes i en felles samling. Jeg ser også på hva slags informasjon i databasene det er aktuelt å hente ut. På grunnlag av dette vurderer jeg hvordan en slik samling på en hensiktsmessig måte kan lagres og brukes til identifisering av protein-/genforekomster og -interaksjoner i artikkel-sammendrag. Forslagene og argumentene i oppgaven underbygges av en prototype som viser hvordan en samling av flere databaser kan foregå.

2 Bakgrunnsteori

Dette kapittelet gir en introduksjon til bakgrunnsteori som er nødvendig for en god forståelse av resten av oppgaven. Teorien som blir beskrevet ligger i hovedsak innenfor områdene bioinformatikk, informasjonsgjenfinning og tekstmining.

2.1 Bioinformatikk

Lacroix og Critchlow i NCBI [1, 2] har definert bioinformatikk som: ”*Det feltet av vitenskap hvor biologi, datamaskinkunnskap og informasjons teknologi integreres for å forme en enkelt disiplin. Det ultimate målet for feltet er og gjør det mulig å oppnå ny biologisk innsikt og lage et globalt perspektiv fra hvor samlede prinsipper i biologi kan bli iakttatt."*

2.1.1 Biologisk informasjonsgjenfinning

Biologisk informasjonsgjenfinning går ut på å gjenfinne informasjon innenfor feltet biologi. Fokuset i denne sammenheng er gjenfinning av protein-/gennavn i artikkelsammendrag, men det finnes også et stort antall andre områder innenfor biologisk informasjonsgjenfinning.

For at identifiseringen av protein-/gennavn, synonymer og interaksjoner fra artikkelsammendrag skal fungere på en god måte må sammendragene forbehandles. Denne forbehandlingen kan i følge McNaught og Ananiadou [3] deles inn i betraktning av ord (leksikalsk nivå), organiseringen av grupper med ord i setninger, som fraser eller paragrafer, (syntaktisk nivå) og meningen som kan bli lagt til disse entitetene på innholdsnivå (semantisk nivå). Ved bruk av ordbok for identifisering av protein-/gennavn og interaksjoner vil leksikalsk nivå være det viktigste. Grunnen til dette er at det ved bruk av en liste med protein-/gennavn for identifisering ikke er noe poeng å gruppere ord eller tillegge dem mening, som er noe av det som foregår på syntaktisk og semantisk nivå.

For at identifisering av protein-/genforekomster skal gi gode resultater må også samling med protein-/genoppføringer forbehandles i samsvar med artikkel-sammendragene. Her beskrives de vanligste stegene i forbehandling av biomedisinsk tekst. Hvilke av disse stegene som er aktuelle kommer an på hva

slags metode som brukes for selve identifiseringen. Dette diskuteres nærmere i kapittel 5.2

2.1.1.1 Grunnleggende tekstoperasjoner

Apostrofer

En del ord ender med apostrof og en sekvens på en eller flere bokstaver. For eksempel *IL-10's*. Da er det behov for å skille hovedordet fra apostrofen og etterhenget fordi det angir et, i språklig hensende, meningsfullt forhold mellom to epiteter[1].[3] Ved identifisering av protein-/genforekomster er dette viktig å ta med i betraktningen fordi protein-/gennavn i større grad en vanlig tekst består av slike typer tegn.

Bindestrek

Bruken av bindestrek er ofte ikke konsekvent og det er ikke alltid klart hvorvidt en tekstanalyserer skal returnere en eller flere ord for slike ord. [3] Som med apostrofer, brukes bindestrek i større grad og på en annen måte i protein-/gennavn enn i vanlig tekst. Det er derfor viktig å vurdere nøye hva slags konsekvenser det vil får for identifiseringen å fjerne bindestreker. Det er ikke minst viktig at dette gjøres konsekvent både i ordboka og i artikkelsammendragene.

Flere formater

Nummer kan forkomme i flere formater og inneholde tvetydige separatorer. Gjelder i tillegg til ordinære tall også telefonnummer, datoer, adresser osv.[3]. Proteinnavn med tall kan forekomme i forskjellige formater i de forskjellige protein-/gen-databasene og i et abstrakt. Dette kan gjøre identifiseringen vanskelig og bør derfor samordnes.

Setningsgrense påvisning

Setninger er vanligvis avgrenset av typiske setningsgrense-merker, som for eksempel punktum, utropstegn og spørsmålstegn. Noen ganger derimot blir andre symboler brukt slik som komma, kolon og semikolon.[3] En stor del av protein-/gen-

[1] Epitet: epite´t epi´teton et fl. epi´teta gr. karakteriserende adjektiv el. lignende tillegg til et substantiv. [4]

navn inneholder tegn som punktum, komma, kolon, semikolon og apostrof. Det er derfor viktig å vurdere konsekvensene dersom man fjerner noen av disse.

2.1.1.2 Morfologisk analyse

Hensikten med morfologisk analyse er å linke heterogene overflatevarianter av leksikalsk analyse til sin rette basisform. De fleste leksikalske elementer med innhold gjennomgår en bøyning. De få som ikke gjør det er for eksempel preposisjoner, adjektiv og konjunksjoner. [3] En morfologisk analyse for protein-/gennavn og -synonymer fra de forskjellige databasene vil ikke være nødvendig, fordi disse allerede er på normal form. Det som derimot er viktig for god identifisering er at det i artikkelsammendragene, der man skal identifisere protein-/gennavn, har gjennomgått en morfologisk analyse. Dette vil føre til at protein-/genforekomstene er mest mulig like de som finnes i ordboka.

2.1.1.3 Stoppord

Stoppord har i tradisjonell gjenfinning liten betydning for gjenfinningsprosessen og bør derfor fjernes fra artikkelsammendragene. Dette er ord som finnes svært ofte i en tekst og er derfor ikke gode termer for å skjelne mellom dokumenter. En annen fordel med eliminering av stoppord, er at det ved en indeksering av dokumenter, vil bli en mye mindre og raskere indeks. Stoppord er ofte artikler, preposisjoner og konjunksjoner.[5] Ved identifisering av protein-/genforekomster i artikkel-sammendrag vil, i motsetning til i tradisjonell informasjonsgjenfinning, fjerning av stoppord føre til dårligere gjenfinning. Dette blir vurdert nærmere i kapittel 4.2.3.1.

2.1.1.4 Forkortelser i biomedisinsk tekst

Forkortelser ender ofte med punktum og kan hindre påvisning av setnings grenser. Forkortelsene kan også være akronymer[2] for termer. For eksempel er APL akronym for acute promyelocytic leukemia og antihospholipid syndrome, og må derfor kobles til den fulle termen.

Utfordringen med å identifisere forkortelser i fri tekst kan deles inn i to hoveddeler. For det første må man finne en liste med kandidater i form av <forkortelse, lang-form>-par basert på parenteser. Dessuten må man skjelne mellom faktiske

[2] Akronym: akrony´m et gr. ord som dannes av forbokstavene (el. flere av de første bokstavene) i flere ord som følger etter hverandre , for eksempel Benelux, NATO (el. Nato), moped [4]

forkortelser og andre parentesforklaringer. Metoder som henter ut kandidater bruker en av to mønstre. Lang form (kort form), som gjelder hvis det kun er ett ord inni parentesen eller kort form (lang form), som gjelder hvis det er flere ord inni parentesen.

Etter at kandidatene er identifisert må algoritmen finne ut hvorvidt de er forkortelser. Dette kan gjøres på flere forskjellige måter og hvilken metode som er best kommer an på typen forkortelser man har med å gjøre. Det finnes databaser med biomedisinske forkortelser, som kan være til hjelp. Noen eksempler på slike ordbøker er AcroMed, ARGH, Standford Biomedical Abberivation Database og SaRaD.[3].

Identifisering av forkortelser i abstrakter er viktig for å forbedre andelen identifiserte proteiner i forhold til de som finnes i abstraktene. I denne oppgaven er det ikke tatt hensyn til forkortelser, siden dette vil være på siden av hovedfokuset. Der er likevel viktig å nevne dette fordi det kan være en utvidelsesmulighet som vil forbedre prototypen. Dette blir diskutert nærmere i kapittel 7.

2.1.1.5 Andre utfordringer

Variasjoner

Protein- og gennavn viser en høy grad av variasjon i litteraturen, inkludert tegn-nivå variasjoner, ord-nivå variasjoner, syntaktiske variasjoner og variasjoner med forkortelser.

Tabell 1 viser eksempler på slike variasjoner.

Tegn-nivå variasjoner	D(2) / D2 SYT4 / SYT IV IGA / IG alpha S-receptor kinase / S receptor kinase Thioredoxin h-type1 eler Thioredoxin h(THL1)
Ord-nivå variasjoner	RNase P Protein / RNase P Interleukin-1 beta precursor / INTERLEUKIN 1-beta PROTEIN / INTERLEUKIN 1 beta Transcription indermeditary factor-2 / trascriptional intermediate factor 2 Hepatic microsomes / liver microsomes
Ord-rekkefølge variasjoner	Collagen type XIII alpha 1 / Alpha 1 type XIII collagen Integrin alpha 4 / alpha4 intergrin

Tabell 1 Eksempel på variasjoner i protein-/gennavn[5]

Navn for nylig oppdagede gener og proteiner

Nye protein-/gennavn blir jevnlig tilført protein-/gendatabasene. Dette kan være navn som er endret eller det kan være nye protein/gen. Disse navnene kan være motstandsdyktige for identifikasjonsmetoder, som kun bruker en liste med protein-/gennavn for identifisering. Derfor må det utvikles regler og modeller som gjenkjenner ukjente protein-/gennavn med deres vanlige karakteristikk.[3]

Tvetydige navn

Variasjoner og nye gen/protein i tillegg til tvetydige navn, som kan bli forvekslet med vanlige Engelske ord, er noen av hovedfokusområdene i denne oppgaven. Disse er alle viktige utfordringer, som kan bidra til å forbedre identifiseringen av protein-/genforekomster i artikkelsammendrag, hvis de løses på en hensiktsmessig måte.

2.2 Informasjonsgjenfinning

Det finnes tre hovedtilnærmelser innen informasjonsgjenfinning. Disse er ordbok-basert, regel-basert og maskin-læring tilnærmelser. I tillegg finnes statistisk og hybrid tilnærmelser. Statistisk tilnærmelse er basert på forskjellig statistisk distribusjon av gruppering in tekst, mens hybrid tilnærmelse kombinerer flere teknikker for termgjenkjenning.[3]. I dette tilfellet er det ordbok-basert tilnærmelse som blir brukt, men de andre metodene er verdt å nevne fordi disse også er mye brukt. Det kan også være aktuelt fordi det i en videreutvikling av forslaget som

utledes i denne oppgaven kan være aktuelt å bruke en kombinasjon av flere av disse.

2.2.1 Ordbok-basert tilnærmelse

Ordbok-basert tilnærmelse bruker en liste med protein-/gentermer for å identifisere protein-/genforekomster i en tekst. Dette gjøres vanligvis ved hjelp av forskjellige del-streng sammenlikningsteknikker. Dette er mer nøyaktig enn regle-basert tilnærmelse (kapittel 2.2.2).

For å lokalisere forekomster i en tekst, brukes eksisterende terminologiske ressurser. Hver sekvens av ord i en tekst som tilsvarer en oppføring i en terminologisk ressurs blir ansett som en termforekomst og bare slike strenger er behandlet som termer. Siden det er en stor mengde nyopprettede, samt forskjellige variasjoner av termer, vil mange termforekomster ikke bli gjenkjent i en tekst hvis man bruker ordbokoppslag rett fram. Dette gjør at sensitiviteten til slike systemer kan bli redusert. For å løse dette kombineres ordbøker med tilleggsprosessering, for å gi bred term gjenkjenning. Eksempler på dette er ”edit-distance[3] operasjoner”, for å implementere en mer fleksible ”string maching” mot ordboken.[3]

2.2.2 Regle-basert tilnærmelse

Regle-basert tilnærmelse er systemer basert på håndskrevne regler. Reglene blir utledet av eksperter. Regel-baserte informasjonsuthentings-system bruker innformasjon i tekstlig form, for eksempel gjennom ordbokoppslag.

Generelt sett går regel-basert tilnærmelse ut på å utvikle regler som beskriver en vanlig navnestruktur for en gitt termklasse. Dette gjøres ved å bruke enten ortografiske, leksikalske eller mer komplekse morfo-gramatiske trekk. Hver sekvens av ord i tekst, som kan bli beskrevet av en regel, blir sett på som en termforekomst. [6]

2.2.3 Maskin-læring tilnærmelse

Maskin-læring er en fullstendig automatisert prosess for kunnskapsuthenting. For at en maskin-lærings tilnærmelse skal fungere trenger man en ekspertannotert-

[3] Edit-distance: Avstanden mellom to tekststrenger er minimumet tegn, som kan settes inn, slettes og erstattes, for å gjøre dem like. [5]

treningssamling, som automatisk utlede identifikasjonsreglene i form av forskjellige statistiske algoritmer. [3]

Maskin-læringstilnærmelse er vanligvis designet for spesielle klasser av entiteter. Tilnærmelsene bruker treningsdata for å lære trekk som er relevante for term-gjenkjenning og klassifisering. Hver sekvens av ord i en tekst som oppfyller kriteriene, basert på lærings-reglene, blir ansett som en term forekomst. Den største utfordringen ved denne tilnærmingen er å velge et sett med representasjons trekk som kan bli brukt for nøyaktig gjenkjenning og klassifisering av term tilfelle. En annen utfordring er påvisning av termsammenhenger av flerordstermer som er mest vanskelig å lære. Dessuten er det en utfordring at forekomsen av pålitelige treningsressurser er veldig liten. [7]

2.2.4 Evaluering

2.2.4.1 Presisjon og recall

Resultater av identifisering av protein-/gennavn, synonymer og interaksjoner i abstrakter kan måles i presisjon og recall. For å oppnå gode resultater for presisjon og recall er det viktig å ha en ordbok med mange oppføringer, men det gjelder også å finne en god metode for å identifisere oppføringer i ordboken, til tross for stavingsvarianter og andre variasjoner.

I denne oppgaven blir det ikke foretatt noen målinger av recall og presisjon i forhold til identifisering av protein-/genforekomster i artikkelsammendrag. Det er likevel viktig å ha disse målene i tankene ved utvikling av en felles ordbok, fordi den har mye å si for god presisjon og recall.

Presisjon

Presisjon er definert som andelen av identifiserte proteiner som er relevante, i forhold til alle som er identifisert.[5] For eksempel når systemet har identifisert proteiner i abstraktene har den funnet 20 proteiner. Men kun 15 av disse er faktisk proteiner vil presisjonen være 15/20=0,75, altså 75 %. Det kan også skje at ord som ikke er proteiner identifiseres, noe som kan komme av at identifiserings-metoden, som blir brukt, omfavner for mye.

Recall

Recall er andelen av relevante proteiner som er identifisert i forhold til alle som finnes i artikkelsammendragene.[5] For eksempel hvis systemet har funnet 25 proteiner i artikkelsammendragene, men det egentlig finnes 30 proteiner vil recall bli 25/30=0,83, altså 83 %. Det kan skje at protein/gener ikke bli identifisert selv om de finnes i artikkelsammendragene. Dette kan komme av at protein/genet ikke finnes i ordboken eller av at det er en variant av protein-/gennavnet systemet ikke klarer å identifisere.

2.3 Filformater

Databasene som brukes til å hente ut protein-/gennavn, -synonymer og -interaksjoner for identifisering i Medline abstrakter, kan lastes ned i forskjellige filformater. Dette er forskjellig fra database til database og kan også variere for hva slags type informasjon man skal hente ut av databasen. De vanligste og mest aktuelle filformatene, i denne sammenheng, er Extensible Markup Language**(**XML) og flat fil.

2.3.1 Extensible Markup Language(XML)

XML ble utformet av en XML arbeids gruppe opprettet av World Wide Web Consortium(W3C) i 1996. XML er en del av SGML[4] som beskriver et dokument fullstendig. XML beskriver en klasse med dataobjekter kalt XML-dokumenter og oppførselen til programmer som prosesserer dem. Et XML-dokument er bygd opp av enheter kalt entiteter, som enten inneholder data som er analysert eller data som ikke er det. Formateringen er bygd opp av tegn som former tegndata og koder en beskrivelse av dokumentets lagrings plan og logisk struktur. XML tilbyr en mekanisme for å pålegge regler for lagrings plan og logisk struktur.[8]

2.3.2 Flat fil

En flat fil er en tekstfil som kun kan bli lest eller skrevet til sekvensielt. De kan inneholde en eller flere oppføringer. En oppføring kan inneholde et eller flere felt. Disse kan være delt med tabulator, komma eller vertikalstrek. De kan også ha to-

[4] 2Standard Generalized Markup Language (SGML) er et metaspråk som kan brukes til å definere formateringsspråk for dokumenter. [8]

bokstav koder som identifiserer de forskjellige feltene. Et felt i en oppføring kan inneholde en data verdi eller være tom. Oppføringer i flate filer inneholder ikke strukturerte relasjoner. [9]

ID IL10_BUBCA Reviewed; 179 AA.
AC Q2PE73;
DT 31-OCT-2006, integrated into UniProtKB/Swiss-Prot.
DT 07-FEB-2006, sequence version 1.
DT 28-NOV-2006, entry version 8.
DE Interleukin-10 precursor (IL-10) (Cytokine synthesis inhibitory
DE factor) (CSIF).

Tabell 2 Eksempel fra UniProtKB/Swiss-Prot lasta ned i flat filformat

Tabell 2 viser at forskjellige felt i en flat fil har to bokstaver som identifiserer dem. Protein-/gennavnet identifiseres for eksempel med bokstavene DE og har synonymer i parentes.

2.4 Relasjonsdatabaser

En relasjonsdatabase er en database som retter seg etter relasjonsmodellen og refererer til databasens struktur og hvordan data er organisert. Relasjonsmodellen er en modell basert på predikativ[5] logikk og sett-teori.[10] En relasjonsdatabase er i følge Date [10] et system hvor data er oppfattet av bruker som tabeller.

En tabell i en relasjonsdatabase består av et sett med rader og kolonner. Hver rad består av et sett med kolonner med en verdi i hver. Alle rader består av samme antall kolonner, men ikke alle kolonnene behøver å ha noe innhold. De kan også ha

[5] Predikativ: [pre´- el. -ti´v] et lat. predikatsord, ord som sammen med et uselvstendig verb danner predikatet (verbalet) i en setning (f.eks.*han er konge, arbeidet synes lett*) (subjektspredikativ), el. i en underliggende setning (f.eks.*hun skrev brevet ferdig (= brevet ble ferdig)*) (objektspredikativ) [4]

verdien NULL, noe som betyr at de ikke har blitt tilegnet noe innhold. En tabell i en relasjonsdatabase bør ha en primær nøkkel. På den måten vil en oppføring være unik og man slipper problemer ved for eksempel oppdatering og man har to like rader og dermed ikke vet hvilken som skal oppdateres. En persons navn er ikke egnet som primærnøkkel, fordi en person kan forandre navn, for eksempel hvis han/hun gifter seg. Det forekommer også svært ofte at mange mennesker har det samme navnet. Personnummer er en mer egnet primærnøkkel i forbindelse med en tabell med personopplysninger, fordi personnummeret er unikt for hvert menneske. Det finnes også muligheter for at flere kolonner i en tabell til sammen kan utgjøre en primærnøkkel.

For å få tilgang til og manipulere data i en relasjonsdatabase blir SQL brukt. SQL er et standard datamaskinspråk for å få tilgang til og manipulere databasesystemer. SQL kommandoer blir brukt for å legge til, fjerne, gjenfinne og oppdatere data i en database.

Tabell: Person

Navn	Firma	E-post
Ola Nordmann	NTNU	o.n@ntnu.no
Bill Gates	Microsoft	bill@microsoft.com
Kari Lien	NTNU	k.l@ntnu.no

Figur 1 Et eksempel på en tabell med navn Person i en relasjonsdatabase

For eksempel kan man foreta SQL spørringen ”SELECT * FROM Person WHERE Firma=`NTNU`”, som henter ut all informasjon om personer som jobber på NTNU fra tabellen Person, i Figur 1. [10]

2.5 Indeksering

Indeksering defineres av Chen m.fl. [11] som prosessen å hente ut termer fra et kontrollert vokabular. Man ender da opp med samling av foretrukne termer som blir brukt til å assistere mer nøyaktig gjenfinning av innhold.

Tre av hovedindekseringstypene er inverterte filer, suffiks-tabell/tre og signatur-filer. Alle de tre indekseringstypene blir omtalt her for å gi en oversikt over hvilke som

finnes og hvordan de fungerer. Dette gir grunnlag for vurderingen av hvilken indekseringsmåte som er best egnet i denne sammenheng. Dette blir diskuter i kapittel 4.2.4.3.

2.5.1 Inverterte filer

Inverterte filer er en ord-orientert mekanisme for indeksering, der formålet er å gjøre gjenfinning mer effektivt og redusere størrelsen på plassen som trengs for å lagre forekomstene av alle indekstermene. Strukturen er basert på to elementer. Vokabular, som er listen med alle de forskjellige indekstermene i teksten, og forekomstene, som er en liste med alle posisjonene for hvor termen forekommer i teksten. For hver term i vokabularet er det pekere til deres forekomst i teksten, lagret i forekomst lista.

Inverterte filer er effektive fordi man ikke trenger å gå gjennom hele tekst dokument for hvert ord i en spørring. Indekseringsteknikken er derimot ikke effektiv når det gjelder å løse kontekst spørringer.[5] En tilnærmelse til inverterte filer som er aktuelt å bruke i denne sammenheng er Lucene.

2.5.1.1 Lucene

Lucene bruker inverterte filer med formål å gjøre søk raskere. Lucene er et fulltekst søkemotorbibliotek, skrevet i Java[6]. Det kan tilpasses den samlingen man ønsker å søke i. Det er aktuelt å bruke Lucene til å indeksere protein-/gennavn fordi indeksen både kan lagres på disk og i minnet, og derfor gir raskt søk. Den gir rom for store datamengder og muligheter for å slå sammen flere indekser. En annen fordel er at den kan indeksere både fra flat fil og database, noe som gjør at man har stor valgfrihet.

Hver oppføring i en indeks inneholder en eller flere felt. Lucene tilbyr fire forskjellige typer felt:

- Nøkkelord: Dette feltet blir ikke forbehandlet, men indeksert og lagret i indeksen ordrett. Dette kan for eksempel være en url, navn, dato, telefon etc.

[6] Java: er et objektorientert programmeringsspråk, utviklet av Sun Microsystems. [12]

- UnIndexed: Blir ikke forbehandlet og ikke indeksert, men er en verdi som er lagret i indeksen som den er. Nyttig for felter som skal vises sammen med resultatet. For eksempel URL eller databasenøkkel.

- UnStored: Det motsatte av UnIndexed. Feltet forbehandles og indeksers men blir ikke lagret i indeksen. Egner seg for å lagre store mengder tekst som ikke trenger å bli mottatt i sin originale form, slik som kroppen til websider, eller andre typer tekstdokumenter.

- Text: Blir forbehandlet og indeksert. Felt av denne typen kan bli søkt mot. Men man må være forsiktig med størrelsen. Hvis dataene som er indeksert er en tekst streng blir den lagret, ellers blir den det ikke.[13]

2.5.2 Suffiks-tabeller og -trær

Suffiks-tabell/tre er indekserings mekanismer som er effektive for søk som foregår innenfor et begrenset område, for eksempel en frase, og andre komplekse applikasjoner som søk i genetiske databaser, men kan også brukes til ord-baserte spørringer.

Suffiks-tabell/tre går ut på å behandle tekst som en sekvens med bokstaver istedenfor en sekvens med uavhengige ord. Suffikser er strenger som går fra forskjellige start posisjoner til enden av teksten.

Suffiks-tabeller/trær er effektive for søk, spesielt for komplekse spørringer, men de er også ressurskrevende å bygge, ikke spesielt egnet for store tekster og er ikke passende for tilnærmede tekst søk. [5]

2.5.3 Signatur-filer

Signatur-filer er en ordorientert indeksstruktur som bruker hash-tabell og bit-masker. De består i hovedsak av to komponenter, en hash-tabell over signaturer og en tekst-signatur-fil. En fordel med signatur-filer, er at de passer for veldig mye tekst, samtidig er det en bakdel at det er mulig å få treff selv om ordet forespurt ikke er i blokken for den passende bit masken.[5]

2.6 Medline

Medline er U.S. National Library of Medicine sin fremste bibliografiske database og innholder over 15 millioner referanser til journalartikler inne biovitenskap, med en konsentrasjon på biomedisin. Kilden til referansene er på nåværende tidspunkt ca 5000 journaler på 37 språk fra hele verden. Siden 2005 har mellom 2000 og 4000 referanser blitt lagt til hver dag. De aller fleste publikasjonene dekket av Medline er forsknings journaler, men det finnes også et lite antall aviser, magasiner og nyhetsbrev, som blir sett på som nyttige innenfor spesielle områder. For referanser lagt til mellom 2000 og 2005 er ca 47 % publisert i USA, ca 90 % er på Engelsk og 79 % har engelske artikkelsammendrag skrevet av forfatterne av artiklene.[14] Det er i disse artikkelsammendragene(fra her av omtalt som abstrakter)protein-/gen-forekomstene identifiseres. Dette for å avgjøre hvorvidt en artikkel er relevant eller ikke for brukeren.

2.7 Databaser for protein/gen

Det finnes svært mange annoterings- og interaksjonsdatabaser som inneholder forskjellig biologisk informasjon. Ikke alle av disse er egnet for identifisering av protein-/genforekomster i abstrakter fordi de enten ikke inneholder tilstrekkelig informasjon eller fordi de ikke er åpent tilgjengelige. I dette kapittelet blir annoterings- og interaksjonsdatabasene som er aktuelle i denne sammenheng beskrevet.

2.7.1 UniProtKB/Swiss-Prot

UniProtKB/Swiss-Prot-samlingen har til hensikt å støtte biologisk forskning ved å vedlikeholde en database med høy kvalitet, som fungerer som en stabil, omfattende, fult klassifisert, rik og nøyaktig annotert protein sekvens kunnskaps-base. Den har omfattende kryssreferanser og spørringsgrensesnitt, som er fritt tilgjengelig. Per dags dato (01.05.2007) inneholder UniProtKB/Swiss-Prot 4.534.260 oppføringer. Databasen tilføres jevnlig nye oppføringer. Bare i perioden 01.04.07-01.05.07 har 312.089 oppføringer blitt oppdatert eller tilført databasen[15]

2.7.2 HUGO Gene Nomenclature Committee (HGNC)

HUGO Gene Nomenclature Committee(HGNC) databasen inneholder over 26.000 gener med synonymer og tidligere navn. Den består hovedsakelig av protein-koding

gener, men inneholder også symboler for pseudogenes, ikke-kodende RNA, fenotyper og genome kjennetegn. Hver oppføring i HGNC har en unik id og symbol. Mange av genen i HGNC inneholder også kryssreferanse til UniProtKB/Swiss-Prot-oppføringer. [16]

2.7.3 IntAct

IntAct er en fritt tilgjengelig proteininteraksjonsdatabase. Interaksjonene er hentet fra litteratur eller direkte brukertilførsel. IntAct er en åpen kilde database og programvare for modellering, lagring og analysering av molekylær interaksjons data. Dataene i databasen stammer i sin helhet fra publisert litteratur og er manuelt annotert av biologi eksperter med et høyt nivå av detalj. Databasen inneholder over 126.000 binære interaksjoner hentet ut fra 2100 vitenskaplige publikasjoner og utnytter omfattende bruk av kontrollert vokabular. Websiden tilbyr verktøy for søk, visualisering og nedlasting av deler eller hele databasen. Interaksjonene i databasen refererer til UniProtKB/Swiss-Prot-oppføringer. [17]

2.7.4 BioGrid

BioGrid er en åpen database med protein og genetiske interaksjoner. Den inneholder over 167.000 interaksjoner fra forskjellige områder innenfor biologien. Et webgrensesnitt tillater søk og på interaksjonsdata. Datasett kan også lastes ned i forskjellige formater, blant annet tabulator-delt tekstfil. [18]

2.7.5 Oversikt over annoterings- og interaksjonsdatabasene

Databasene	Type database	Aktuelle felter	Komprimering	Filformat
UniProtKB/-Swiss-Prot	Protein-/gen-annotrings-database	Primary accession number, Protein name, Synonyms, Modified	gzip	XML, Fasta, Flat File
HGNC	Protein-/gen-annoterings-database	Approved symbol, Approved name, Aliases, Name aliases, Date modified	Ingen komprimering	Text, HTML
IntAct	Interaksjons-database	ID interactor A, ID interactor B	zip	Tab
BioGrid	Interaksjons-database	Official symbol A, Official symbol B	zip	Tab

Tabell 3 Oversikt over de aktuelle databasene

Tabell 3 gir en oversikt over de aktuelle databasene og hva slags type database de er. Kolonnen *Aktuelle felter* gir en oversikt over de aktuelle feltene fra hver enkelt database. Disse feltene blir nærmere omtalt i kapittel 4.2.2. Kolonnene *Komprimering* og *Filformat* gir en oversikt over hva slags komprimeringstype filene har når de lastes ned og hvilke aktuelle filformat som finnes.

3 Tidligere arbeid

Flere, blant annet Moon og Singh [19], påpeker at en felles database for protein-/genforekomster og -interaksjoner hadde vært nyttig og at mye informasjon er for dårlig tilgjengelig. Det er likevel svært lite forskning tilgjengelig på dette området. De fleste som har brukt databaser/ordbøker til å gjenfinne protein-/gen-forekomster og -interaksjoner i abstrakter har ofte kun brukt protein-/gennavn fra en enkelt database. Dessuten er det mange tilnærmelser som ikke tar i bruk slike databaser overhode. Det er derfor ikke bare viktig å se på hva som er gjort innen området når det gjelder å lage en felles ordbok, men også hvordan protein-/gen-forekomster blir identifisert. På den måten kan tilnærmelsen til en felles ordbok tilpasses de behovene som finnes innen protein-/genidentifisering i artikkel-sammendrag. I det følgende presenteres derfor noe av den nyeste tilgjengelige forskningen innenfor området identifisering av protein-/genforekomster og -interaksjoner i abstrakter.

Som omtalt i kapittel 2.2 finnes det hovedsakelig tre tilnærmelser til protein-/gen-identifisering i artikkelsammendrag, ordbok-basert, regel-basert og maskin-læring. Ordbok-basert tilnærmelse er det området der det, i utgangspunktet, er mest aktuelt å ta i bruk en samling bygd opp av flere annoterings- og interaksjons-databaser, fordi det der identifiseres protein-/genforekomster på grunnlag av en liste med protein-/gennavn.

3.1 Ordbok-basert tilnærmelse

Yamamotoy m.fl. [20] påpeker at tidligere tilnærmelser innen for håndlagde regler og maskin-læring ignorerer det faktumet at biologiskeoppføringer har grensetvetydigheter. Ulikt generell engelsk, er ikke mellomrom et egnet tegn for avgrensning. Dette blir forklart med at navnebeskrivelser i biomedisinske ressurser for det meste er sammensetninger. For å løse dette problemet foreslås det av Yamamotoy m.fl. [20] en grammatisk analyse som oppnår sofistikert tolkning og tilpasser seg til biomedisinske resurser effektivt. En annen utfordring innen bioinformatikk er stoppordliste, siden man ikke kan bruke en tilsvarende som ved

tradisjonell informasjonsgjenfinning. Moon og Rahul [19] har påpekt at enkel, spesialtilpasset stoppordliste er hensiktsmessig for informasjongjenfinning i dette domenet, men en stor domenespesifikk ordbok vil sannsynligvis fungere bedre. Det samme gjelder for stemming.

Tsuruokazy og Tsujiiyz [21] har også identifisert flere utfordringer ved ordbok-basert tilnærmelse. For det første forekommer et stort antall falske gjenkjenninger, hovedsakelig på grunn av korte navn. Dette vil igjen lede til lav presisjon. Dessuten kan man få lav recall på grunn av stavingsvarianter i protein-/gennavn. Tsuruokazy og Tsujiiyz [21] foreslår imidlertid en løsning på disse problemene. For problemet med falske gjenkjenninger på grunn av korte navn, foreslås det å bruke en klassifiserer, som finner ut om hver kandidat er et proteinnavn eller ikke. Lav recall på grunn av stavingsvarianter kan i følge Tsuruokazy og Tsujiiyz [21] løses ved å bruke tilnærmet tekst-streng-likhet metode hvor overflate likheter mellom termer blir tatt i betraktning.

Både Yamamotoy m.fl. [20] og Tsuruokazy og Tsujiiyz [21] påpeker flere aktuelle utfordringer innen ordbok-basert tilnærmelse. Både at biologiske oppføringer har stavingsvarianter og at tegn som brukes i vanlig informasjonsgjenfinning til å skille mellom ord, ikke kan brukes, er viktig å være klar over for å kunne ha muligheten til å utvikle forbedrede ordbok-baserte tilnærmelser. Falske gjenkjenninger er også et av hovedproblemene i følge Tsuruokazy og Tsujiiyz [21]. Selv om dette er store utfordringer er de ikke uløselige. Yamamotoy m.fl. [20] foreslår grammatisk analyse som løsning på utfordringen med at oppføringer har stavingsvarianter. Her er det også muligheter for tilnærmet treff.

Det er viktig å være klar over at biologisk tekst ikke bør forbehandles og indekseres på samme måte som ved vanlig informasjonsgjenfinning. Fjerning av typiske tegn som punktum, komma, apostrof o.l. kan føre til at gjenfinning blir tilnærmet umulig. Dessuten kan en typisk indeksering der hvert ord blir egen oppføring i indeksen, i tillegg til andre faktorer, være kilden til falske gjenkjenninger. Alle disse utfordringene blir diskuter og forsøkt løst på en hensiktsmessig måte i denne oppgaven.

3.2 Regel-basert tilnærmelse

Regel basert læring er en nyttig tilnærming ved forskning på nye måter å finne protein-/genforekomster og -interaksjoner, men maskin-læring er målet for de fleste av disse systemene. Det finnes derfor ikke mye ny forskning innenfor dette området. Mange av de tilgjengelige artiklene skrevet er fra før år 2000. Det finnes imidlertid noen nyere artikler som sier noe om nåværende tilstand innefor området.

Nyere forsking på dette området, foretatt av Kim m.fl.[22], omhandler identifiseringen av proteininteraksjoner. [22] ønsker å hente ut informasjon mellom proteiner for å argumentere mot informasjon i nåværende proteindatabaser som hovedsakelig konsentrere seg om positive fakta for å finne relasjoner. Kim m.fl.[22] argumenterer for at kontrastrelasjoner er nyttige for å utforske det ukjente, siden det kan føre til mer forståelse av observerte fenomener.

Tilnærmelsen til Kim m.fl.[22] går ut på at hver del av kontrastinformasjonen er laget av to deler. Et kontrast par av to eller flere objekter som er i kontrast, kalt fokusobjekter, og en biologisk egenskap eller prosess som kontrasten er basert på, kalt forutsatt egenskap. Metoden som blir brukt går ut på at det i et abstrakt først lokaliseres en setning som inneholder 'not'. Så identifiserer den kontrast utrykk fra disse setningene. Hvis kontrast utrykkene er, eller kan bli redusert til proteinnavn, produserer systemet en kontrast mellom de to proteinene. Systemet analyserer den semantiske likheten mellom de koordinerte uttrykkene ved å analysere ord-nivå likhet mellom utrykkene. Dette gjøres ved hjelp av en Part-Of-Speech(POS)[7] tagger som finner hvert ord sin mest hyppige POS tagg ved å slå opp i en manuelt perseptuell ordbok sammen med noen domenespesifikke korreksjons regler. Det kan også gjøres ved bruk av en substantivfrase gjenkjenner som ser etter substantivfraser som starter eller ender med et forutbestemt ord eller POS-tagg.

Hvis ingen av stegene fungerer prøver systemet å identifisere parallellisme utrykt innenfor setningen. Parallellisme for en kontrast refererer til et par med benektende

[7] Part-of-speech(POS) tagging: Forbehandlings steg som er viktig for å takle de forskjellige formene av orbok-syntaktisk tvetydigheter av ord. [3]

setningsgramatikk og en bekreftende setningsgramatikk som er enten semantisk identiske eller i det minste i en subsumpsjon[8].

For å kunne lage en nyttig kunnskapsbase med proteinkontraster, er det viktig å krysslinke kontrastproteinen-navnene fra litteraturen med oppføringer i en standard protein database som for eksempel UniProtKB/Swiss-Prot. For å kunne gjøre dette har Kim m.fl.[22] utviklet en proteinnavn-grunnmodul som finner tilsvarende UniProtKB/Swiss-Prot-oppføringer for protein-/gennavn. Dette gjøres med manuelt konstruerte mønstre som behandler variasjoner i proteinnavn når det gjelder spesielle tegn, flertall, forkortelser, kasserbare ord og akronymer.

Kim m.fl.[22] argumenterer for at kunnskapen man får ut av dette kan brukes til å skjelne mellom nøyaktige biologiske roller for tilsynelatende like proteiner. Dessuten kan et kontrastforhold vise til likhet eller funksjonelt forhold mellom de fokuserte objektene som er i kontrast. Slik likhetsinformasjon kan være nyttig når man skal forstå det som virker mellom proteinene, spesielt hvis likheten mellom disse proteinene ikke er åpenbare.

En noe eldre artikkel av Narayanaswamy m.fl.[23] har tilnærmet seg området ved å utvikle et symbolsk verktøy basert på et sett med manuelt utviklede regler. Reglene utforsker overflatespor, enkel lingvistikk og domenekunnskap ved å identifisere relevante termer i biologiske artikler. Narayanaswamy m.fl. [23] oppgir flere grunner for å bruke manuelt utviklede regler. Blant annet mangelen på annoterte samlinger. Det finnes heller ikke nøyaktig overensstemmelse om hva som utgjør et navn, selv ikke med begrensninger som protein/gener. Dessuten vil behovet for andre typer navneentiteter oppstå ettersom informasjonsuthentingsjobben går framover. Narayanaswamy m.fl.[23] fokuserer ikke kun på protein og gen navn men også på kjemiskenavn. Grunnene til dette er at kjemiskenavn deler forskjellige kjennetegn og blir brukt til å identifisere protein-/gennavn. Dette kan hjelpe til med å forbedre presisjonen for proteinnavn-identifisering. En annen grunn er at det i biologiske-interaksjoner ikke bare er proteiner og gener involvert, men også kjemikalier. Narayanaswamy m.fl.[23] ønsker med sitt system å hente forskjellige biologiske-termer og klassifisere dem i kategorier.

[8] Subsumpsjon: subsum(p)sjo´n: underordning, sammenfatting under noe, forutsetning [4]

Systemet identifiserer forkortelser, kjernetermer og funksjonelletermer. Det førstnevnte fungerer på det premisset at den første forekomsten av forkortelsen forekommer i et mønster der den originale termen er fulgt av en parentes med forkortelsen. Identifisering av kjerne termer og funksjonelle termer er en basis funksjon.

Sammenkjeding og utvidelsesregler i systemet brukes for å hente ut navn som består av flere ord. Klassifikasjonsmetoden som blir brukt går ut på at det først og fremst ble forsøkt å klassifisere på bakgrunn av termer og suffikser. Suffikser av lengde tre, fire eller fem, blir tatt i betraktning. I en testsamling blir navnene klassifisert først ved å se på det siste ordet. Hvis man får treff lista med termer, blir navnene tagget i forhold til tilhørende klasse. Hvis ikke termen finnes, blir suffiksen til det siste ordet av navnet hentet ut og det blir forsøkt funnet treff i forhold til suffiks lista. Hvis man heller ikke da får noen treff, forblir navnet uklassifisert.

Klassifiseringa av navn kan også foregå ved bruk av likhetseksempler. Det gjøres ved at man for et navn i testsamlingen forsøker å finne ut hvor likt det er kandidater fra treningssamlingen. For hvert par, starter man med delvis treff ved å sammenlikne et og et ord. Legger poeng til hele par av navn. Begynner med å legge posisjons nummer til hvert par av ord. Starter med treffet lengst til høyre, med posisjon 0. Hvis det ikke blir treff til høyre for det første treffet blir den tildelt en negativ posisjon. Narayanaswamy m.fl.[23] påpeker at mulig videre arbeid er å gjøre dette automatisk istedenfor å håndkode termer med vekter for forskjellige parametere.

Kim m.fl.[22] og [23] presenterer to forholdsvis ulike tilnærmelser til identifisering av protein-/genforekomster og -interaksjoner i artikkelsammendrag. Kim m.fl.[22] bruker regler basert på kontraster, mens Narayanaswamy m.fl.[23] utforsker overflatespor, enkel lingvistikk og domenekunnskap ved å identifisere relevante termer. Begge er interessante tilnærmelser, men som tidligere påpek er regel-baserte tilnærmelser som oftest kun et forstadium til utviklingen av en maskin-lærings tilnærmelse. Det betyr ikke at regel-baserte tilnærmelser ikke er viktige. De svært viktig for å kunne forske på nye metoder for å forbedre gjenfinning, som etter hver kan bli automatisert. Identifiseringen av interaksjoner i denne oppgaven vil foregå på grunnlag av allerede utviklede databaser med protein-/geninteraksjoner,

men som man kan se i Kim m.fl.[22] er det fult mulig å bruke regler sammen med ordbok og på den måten forbedre identifiseringen. Dette kan være aktuelt for en videreutvikling av prototypen.

3.3 Maskin-læring tilnærmelse

Mange regel-baserte og ordbok-baserte tilnærmelser videreutvikles til å bli maskin-lærings metoder eller en kombinasjon av flere metoder. Dette er derfor et område det er og har vært forsket mye på. Det finnes flere aktuelle tilnærmelser innenfor maskin-lærings området.

Soni[24]beskriver en proteinnavn-taggingmetode som er et fundamentalt forstadium til uthenting av protein interaksjoner fra Medline abstrakter. Denne metoden behandler forbunnede kjennetegn i forhold til generaliserings karakteristikk fra Support vector machine(SVM)[9]. I likhet med Yamamotoy m.fl. [20] påpeker også Soni [24] at mange tidligere tilnærmelser ignorerer det faktumet at bioentiteter har grense-tvetydigheter. Ulikt generell engelsk, er et mellomrom ikke et tilstrekkelig symbolskilletegn. For å løse problemet foreslås en grammatisk analyse som oppnår sofistikert symbolisering og tilpasser seg til biomedisinske ressurser effektivt. Et problem med tidligere tilnærmelser er at de ikke takler mange innbyrdes forbunnede kjennetegn. Feil i en tidlig fase av regel applikasjoner blir ofte overført til seinere faser og ødelegger den totale ytelsen.

Bunescua m.fl.[25] har utviklet et tekstbrytnings-system som identifiserer protein-/gennavn i tekst ved bruk av et trent Conditional Random Field(CRF) og identifiserer interaksjoner via en filtrert felles-henvisnings analyse. Rapporterer også to nye strategier for identifisering av interaksjoner i tekst ved bruk av innlærte mønster-baserte regler og SVM. En del interaksjons kart(DIP, BIND, HPRD etc.) er laget men selv om interaksjonene i disse datasettene er hentet fra samme kilde, er settene ganske forskjellige fra hverandre. Grunnen til dette er at settene er ensidige for forskjellige klasser av interaksjoner og at antallet interaksjoner i Medline abstrakter er enormt.

[9] Support vector machine(SVM): et sett med relaterte overvåkede læringsmetoder for bruk ved klassifisering og regresjon. [3]

Daraselia m.fl.[26] presenterer i en publisert artikkel MedScan, et komplett infomasjons-uthentingssystem, som tolker den semantiske strukturene ved bruk av pathway-orientert ontologi og henter ut protein funksjon informasjon. MedScan er et tre-lags informasjonsuthentingssystem basert på en full-setning syntaksanslyse-tilnærmelse.

Problemer med systemet er for det første at den syntaktiske naturen til setningene som beskriver interaksjonene, gjør automatisk uthenting til en utfordrende oppgave og kan i beste tilfelle tjene som en praktisk test av MedScan ytelse. For eksempel setninger som ser tilnærma like ut, men som har helt forskjellig betydning. Når dataene ble evaluert var 91 % av interaksjonene riktige. Dessuten var det et problem at feil som oppstod i de siste 8% var feiltolkning av setninger som inneholdt ordet ’interact’ eller ’interaction’. [26]

Det er foretatt mye forskning innen for maskin-læring, både når det gjelder identifisering av protein-/genforekomster og -interaksjoner. De forskjellige systemene bruker ulike tilnærmelser. Selv om dette er mer eller mindre automatiske systemer, kan man se eksempler på bruk av allerede utviklede databaser. For eksempel bruker Bunescua m.fl. [25] interaksjonsdatabasene DIP, BIND, HPRD i sin tilnærmelse. Dette viser et eksempel på at tilnærmelser kan kombineres og dermed kan bedre systemer utvikles. Det er med andre ord viktig å være klar over at selv om en tilnærmelse i utgangspunktet er enten en ordbok-basert, regel-basert eller maskin-basert tilnærmelse, kan den utvikles til å bli en kombinasjon av flere tilnærmelser og dermed bli bedre og mer nyttig i gjenfinning.

4 Egen løsning

4.1 Basis idé

Basis ideen er å forsøke å samle flere annoterings- og interaksjonsdatabaser i en relasjonsdatabase, for å se hvordan dette vil fungere, både når det gjelder selve kombinasjonen av flere databaser og i forhold til identifisering av protein-/gennavn, -synonymer og -interaksjoner.

For å kunne finne ut om det er mulig, på en god måte, å samle flere protein-/gen-databaser i en relasjonsdatabase, har jeg utviklet en prototype som henter ut informasjon fra filer, med databaseinnholdet, lastet ned fra de forskjellige annoterings- og interaksjonsdatabasene. Informasjonen analyseres og legges i en relasjonsdatabase. En indeks bygges fra databasen ved hjelp av Lucene og gjør det mulig å søke i indeksen og å kunne bruke prototypen mot allerede utviklede applikasjoner for protein-/genidentifisering i abstrakter.

4.2 Vurderinger

4.2.1 Protein-/gennavn og -interaksjonsdatabasene

Forutsetningen for å kunne kombinere flere annoteringsdatabaser i en relasjons-database er at det må finnes en form for referanse mellom databasene. Grunnen til dette er at selv om et protein/gen er det samme, skrives det ofte forskjellig i forskjellige annoteringsdatabaser. Dessuten opererer de forskjellige databasen med forskjellige typer identifikasjon av protein-/gennavn. Symbol er unik i UniProtKB/-Swiss-Prot og id er unik i HGNC.

Mange oppføringer i HGNC har en kryssreferanse til en UniProtKB/Swiss-Prot-oppføring, men dette betyr ikke at protein-/gennavnet er tilsvarende i de to annoteringsdatabasene. Et eksempel er proteinet `*basic helix-loop-helix domain containing, class b, 2*` med id `*1046*` i HGNC. Det samme proteinet har symbol `*O14503*` og heter `*class b basic helix-loop-helix protein 2*` i UniProtKB/Swiss-Prot.

Fordi HGNC har en kryssreferanse til UniProtKB/Swiss-Prot-symbolet i oppføringen, kan man se at dette er samme protein-/gennavnet.

Både HGNC og UniProtKB/Swiss-Prot har kryssreferanser til hverandre, men UniProtKB/Swiss-Prot inneholder også kryssreferanser til mange andre annoteringsdatabaser. Dette gir to muligheter for å referere protein-/gennavn til hverandre i relasjonsdatabasen. Referansen kan enten hentes fra UniProtKB/-Swiss-Prot eller fra den databasen som skal legges til. For at relasjonsdatabasen skal inneholde minst mulig informasjon, er det i utgangspunktet best å bruke en referanse mot UniProtKB/Swiss-Prot, fra den annoteringsdatabasen som skal legges. Grunnen til dette er at hvis man skal referere fra UniProtKB/Swiss-Prot må alle oppføringer lagres med alle kryssreferanser. I motsetning vil man kunne lagre kun de brukte kryssreferansene hvis det refereres fra en annen database til UniProtKB/Swiss-Prot. Dette vil gjøre databasen betydelig mindre og raskere, fordi hver oppføringe i UniProtKB/Swiss-Prot inneholder minst 10 kryssreferanser og som tidligere nevnt inneholder UniProtKB/Swiss-Prot 4.534.260 oppføringer[15]. Med mindre veldig mange databaser skal kombineres, blir store deler av denne informasjonen overflødig og tar opp mye plass.

Oppføringer kan også være så tilnærmet like at det er på mange måter er overflødig å lagre dem som to oppføringer. For eksempel `*BCL2 binding component 3*` og `*bcl-2-binding component 3*` der eneste forskjell er to bindestreker. Slike variasjoner i protein-/gennavn kommer, i følge McNaught og Ananiadou [3], ofte av at forfatterne av artiklene er lite konsekvente i sin skrivemåte. Et mulighet er derfor å fjerne bindestrekene i alle oppføringene i relasjonsdatabasen og gjøre det samme i abstraktene. I prototypen utviklet her har jeg valgt ikke å fjerne bindestreker og beholde alle variantene av protein-/gennavnet. Grunnen til dette blir nærmere diskutert i kapittel 4.2.3.1.

Noe annet som forholdsvis ofte forekommer i annoteringsdatabasene er at samme proteinnavn har flere oppføringen med forskjellig id/symbol. Et eksempel fra UniProtKB/Swiss-Prot er `*Class B basic helix-loop-helix protein 2*` som finnes både i oppføringen med symbolene `*O14503*` og `*O35185*`. Dette gjør at begge symbolene må lagres i databasen for å vise at dette er to forskjellige oppføringer med samme navn i UniProtKB/Swiss-Prot.

Til slutt må det nevnes at et protein-/gennavn kan være en egen oppføring, men det kan også være et synonym for et annet protein. Dette er i seg selv ikke noe problem, men det vil føre til en noe dobbeltlagring av protein-/gennavn i relasjonsdatabasen. Det er ikke hensiktsmessig å fjerne disse dobbeltoppføringene fordi disse kan bidra til å finne relasjoner mellom protein-/gennavn.

Oppsummert kan det som er omtalt i dette kapittelet føre til følgende kombinasjonsmuligheter i en relasjonsdatabase med UniProtKB/Swiss-Prot- og HGNC-oppføringer:

- Flere oppføringer kan ha samme protein-/gennavn. Må da skilles på symbol. Dette forekommer forholdsvis ofte i UniProtKB/Swiss-Prot, men ganske sjeldent i HGNC. Det forekommer også hyppig på tvers av databasene.

- Oppføringer i UniProtKB/Swiss-Prot og HGNC kan være tilnærmet like, dvs. det kan bare være en bindestrek eller andre småting som skiller dem. Som tidligere nevnt blir alle versjonene tatt med.

- Flere oppføringer kan ha same synonym. Dette innebærer at flere UniProtKB/Swiss-Prot kan ha samme synonym eller at flere HGNC-oppføringer kan ha samme synonym. Det betyr også at et synonym fra UniProtKB/Swiss-Prot kan være synonym for en eller flere oppføring i HGNC og omvendt.

- Et proteinnavn kan være synonym for en annen oppføring. Dette gjelder både innad i annoteringsdatabasene og på tvers av dem.

Som nevnt er UniProtKB/Swiss-Prot-annoteringsdatabasen et bra utgangspunkt fordi den inneholder mange kryssreferanser til andre databaser. Den er også en forholdsvis omfattende annoteringsdatabase med 4.534.260 oppføringer. UniProtKB/Swiss-Prot var dermed et naturlig valg. Når det gjelder andre annoteringsdatabaser som var aktuelle å bruke, var ikke HGNC den eneste muligheten. En annen annoteringsdatabase som ble vurdert var Protein Data Base(PDB). Dette er en verdensomspennende samling av informasjon om tredimensjonale strukturer for store biologiske makromolekyler. PDB databasen kan man lastes ned i en enkelt fil. Problemet med denne nedlastingen er at den kun inneholder protein-/gennavn og -id. Mer utfyllende informasjon kan hentes ut, men

da må det gjøres for hvert protein via et webgrensesnitt eller nedlasting av enkeltfiler. I forhold til HGNC gir den lite utfyllende informasjon, som for eksempel synonymer. UniProtKB/Swiss-Prot kryssrefererer til PDB, så slik sett er den egnet for bruk i denne sammenheng. Den avgjørende faktoren for valg av HGNC var derfor at mer utfyllende informasjon om hver protein-/genoppføring var lettere tilgjengelig enn i PDB. Dette fører til at HGNC er mer egnet for å teste ut hvordan en samling av flere annoteringsdatabaser vil fungere. For en eventuell utvidelse av relasjonsdatabasen med flere annoterings databaser, vil PDB være aktuell.

To andre annoterings databaser som også ble vurdert, men forkastet var BioThesaurus og STRING. BioThesaurus ble forkastet fordi protein-/gen-oppføringene i den nedlastbare versjonen ikke var tilknyttet noen unik id. STRING var et reelt alternativ fordi den kun innehold synonymer, noe som kunne være nyttig for å tilføre relasjonsdatabasen mer utfyllende informasjon. Det finnes derimot ingen kryssreferanse mellom STRING og UniProtKB/Swiss-Prot-oppføringer, som kan brukes i denne sammenheng.

Når det gjelder interaksjonsdatabaser, er ikke de bygd opp på samme måte som annoteringsdatabasene. Både IntAct og BioGrid inneholder kun referanser til UniProtKB/Swiss-Prot og HGNC. Disse referansene består av par av symboler eller id fra annoteringsdatabasene og det eneste som kan oppstå er at det finnes duplikater av interaksjoner. Dette vil enkelt kunne løses ved å sjekke om interaksjonen allerede finnes i databasen, før den legges til. For at interaksjonsdatabasene skal være til noen nytte i en relasjonsdatabase, må databasen som interaksjonene refererer til, også ligge i databasen. I dette tilfelle har jeg valgt en interaksjonsdatabase som inneholder UniProtKB/Swiss-Prot-symboler og en som inneholder HGNC-id. Samtidig inneholder HGNC-interaksjoner som ikke har HGNC-id. Faktisk er kun 31.533 av de 167.572 oppføringene i databasen interaksjoner der begge protein/gen er HGNC-id. Det ble derfor valgt å fjerne alle oppføringer der ikke begge protein/gen var HGNC-id og bruke resten som testdatabase.

4.2.2 Databaser

4.2.2.1 UniProtKB/Swiss-Prot

For dette formålet var det aktuelt å laste ned UniProtKB/Swiss-Prot i enten XML- eller flat filformat. XML-fil kan enkelt analyseres, ved bruk av ferdige klasser i java, er derfor et naturlig valg. Fra flat filformat vil det kreve mer å få ut informasjonen man trenger, samtidig som man også vil få ut informasjon fra fila som ikke skal brukes. Ved å bruke XML kan kun de feltene som skal brukes, hentes ut. UniProtKB/Swiss-Prot i XML-format er ca 2 gigabyte.

UniProtKB/Swiss-Prot.xml inneholder symbol, protein-/gennavn, -synonymer, opprinnelsen til protein/genet, nøkkelord, kryssreferanser, samt flere andre felt. Tillegg B, viser et eksempel på en oppføring fra UniProtKB/Swiss-Prot i XML-format.

For å kunne identifisere protein-/genforekomster fra abstraktene trenger man i utgangspunktet kun protein-/gennavnet, *Protein name* feltet, fra UniProtKB/-Swiss-Prot. Men dette feltet alene vil ikke kunne gi en relasjonsdatabase. For at relasjonsdatabasen skal ha mer utfyllende informasjon med synonymer og interaksjoner samt relasjon til andre annoteringsdatabaser, må enten feltet *Primary accession number* eller *Entry Name* også legges inn i *ProtNavn*-tabellen sammen *Protein name*. Både *Primary accession number* og *Entry name* er i seg selv en unik id for en oppføring, man trenger derfor ikke begge. Det som avgjør hvilket av disse to feltene som skal brukes er interaksjonsdatabasene. To av de aktuelle består av interaksjoner utrykt både ved *Primary accession number* og *Entry name*. Problemet med *Entry name* er at det ikke er skrevet fult ut. For eksempel `*104K_THEAN*` er skrevet `*104k_(protein)*` og vil derfor ikke være unikt. Dessuten er ikke *Entry name* like stabilt som *Primary accession number* i følge UniProt Consortium [15]. Dette avgjør at *Primary accession number* brukes sammen med *Protein name* i tabellen *ProtNavn* fra denne databasen.

Tabellen *ProtNavn* vil ha relasjoner til tabellene *Synonymer* og *Interaksjon*. En protein-/genoppføring kan ha mange synonymer og være en del av mange interaksjoner. Relasjonen vil skje på *Primary accession number*, som vil finnes i alle de tre tabellene.

Selv om *Primary accession number* er unik for alle oppføringene vil ikke det fungere som primærnøkkel i tabellene. Dette fordi et protein-/gennavn i relasjonsdatabasen kan finnes i både UniProtKB/Swiss-Prot, HGNC og kan også forekomme flere ganger i samme annoteringsdatabase. Dermed må alle symbol/id-er fra begge annoteringsdatabasen være med for at man skal kunne relatere synonymer og interaksjoner til protein-/gennavn oppføringene fra både UniProtKB/Swiss-Prot, HGNC og innad i samme annoteringsdatabase.

Et annet felt fra UniProtKB/Swiss-Prot som er aktuelt er *Synonyms*. Synonymene vil sammen med protein-/gennavnen kunne hjelpe til med å identifisere protein-/gen-forekomster i abstrakter. Et synonym vil være tilknyttet et eller flere proteinnavn ved hjelp av *Primary accession number*.

4.2.2.2 HUGO Gene Nomenclature

Annoteringsdatabasen kan lastes ned i HTML eller ren tekst. Hver oppføring har en unik id og 18.089 av 24.381 oppføringer [16] har referanse til UniProtKB/Swiss-Prot. Ved å bruke denne databasen sammen med UniProtKB/Swiss-Prot kan man bygge opp en relasjonsdatabase med synonymer, som kan fungere bra ved identifisering av protein-/gennavn og -interaksjoner. Ved tilpasset nedlasting kan man velge hvilken informasjon som skal være med. Informasjonen som er nyttige å legge i databasen er hovedsakelig protein-/gennavn, id, synonymer og UniProtKB/Swiss-Prot-referanse. Av synonymer finnes to typer, navne synonymer og synonymer som ikke er direkte knyttet til navnet. Disse blir begge behandlet som synonymer og ikke skilt mellom i relasjonsdatabasen. Grunnen til dette er at kun 1.219 av 24.381 oppføringer har navne synonymer og det vil dermed ikke være noen grunn til å gjøre relasjonsdatabasen mer komplisert ved å skille mellom dem. Ved bruk av andre eller flere annoteringsdatabaser, kan dette likevel bli aktuelt.

HGNC kan lastes ned i html eller tabulator-delt tekst. Ingen av disse formatene er spesielt godt egnet for å plukke ut den informasjonen som skal i relasjons-databasen. Det er vanskelig å identifisere og hente ut informasjon fra en ren html-fil fordi html-taggene den ikke gir noen pekepinne på innholdet. I en tabulator delt tekstfil vil derimot enklere kunne identifisere informasjonene man er ute etter ved at hver kolonne er skilt med tabulator. Det vil derfor i denne sammenheng være mest

hensiktsmessig å bruke tabulator-delt tekstfil. Tillegg B, viser eksempler på oppføringer fra HGNC i tabulatordelt-delt tekst.

Som i UniProtKB/Swiss-Prot trenger man en id i tillegg til protein-/gennavnet for å kunne ha relasjoner til synonymer og interaksjoner. Også her finnes det to forskjellige unike id-er. *HGNC ID* er en unik id, mens *Approved Symbol* er det offisielle gen symbolet godkjent av HGNC[16]. I dette tilfelle velges *HGNC ID* fordi den er unik[16]. Dessuten består interaksjonsdatabasen BioGrid av par med HGNC-id. Derfor vil denne id-en som ligger i *ProtNavn*-tabellen ha en eller flere relasjoner med *Interaksjon*-tabellen. Den vil også være relatert til *Synonymer*-tabellen.

Approved name kalles feltet som inneholder protein-/gennavnet som brukes til å identifisere protein-/genforekomster og -interaksjoner i abstrakter. Det vil stå som primærnøkkel i tabellen *ProtNavn* sammen med protein-/gennavnene fra UniProtKB/Swiss-Prot.

Som UniProtKB/Swiss-Prot inneholder også HGNC-synonymer. Det er to typer synonymer, *Aliases* og *Name aliases*. Begge disse feltene kan inngå i *Synonymer*-tabellen og vil kunne føre til at flere protein-/genforekomster identifiseres i abstrakter. Dermed vil recall kunne forbedres.

4.2.2.3 IntAct

IntAct databasen kan lastes ned i flere formater. Blant annet som tabulator-delt tekstfil, som vil være mest egnet i dette tilfellet. Den består av mye forskjellig informasjon. Det er kun kolonnene som utgjør par av interaksjons symboler som er nyttig i denne sammenheng. Noen eksempler på oppføringer fra IntAct i tabulator-delt fil, vises i tillegg C.

Databasen inneholder ikke protein-/gennavn, men to felt med samhandlende Primary accession number fra UniProtKB/Swiss-Prot. Feltene heter henholdsvis *ID interactor A* og *ID interactor B*. Disse kan brukes til å finne de aktuelle protein-/gen-navnene i *ProtNavn*-tabellen og man må derfor ha en relasjon mellom *Interaksjon* og *ProtNavn*-tabellen. Databasen inneholder flere felter enn UniProtKB/Swiss-Prot *Primary Accession Number*. Blant annet *Entry name*. Et eksempel på et *Entry name* er *104K_(protein)*. Som man ser er ikke dette feltet fullstendig, fordi (protein) kan

erstattes med mange forskjellige proteinnavn, og vil derfor ikke være unikt. Dette vil føre til at det ikke kan brukes mot tabellen *Protein*. Alle UniProtKB/Swiss-Prot-interaksjonene starter med frasen *uniprotkb*. Det finnes også oppføringer som ikke begynner på denne måten. Disse vil bli fjernet under forbehandlingen. Se kapittel 5.2.

4.2.2.4 BioGrid

Denne databasen kan, i likhet med IntAct, lastes ned i tabulator-delt tekst. Dette i tillegg til at strukturen i filene er omtrent den samme, gjør at samme metode kan brukes for å hente ut data fra begge interaksjonsdatabasene. Eksempler på oppføringer fra BioGrid finnes i tillegg C.

Som IntAct består den nedlastede fila av mye informasjon som ikke vil være nyttig i denne sammenheng. De to første feltene i databasen er *Interactor_A* og *Inteactor_B*. Disse feltene inneholder sammen par av interaksjons protein/gen i form av *HGNC ID*. Interaksjonene i form av HGNC-symboler finnes også. *HGNC ID* blir bruk fordi den som tidligere nevnt er unik.Denne databasen inneholder i tillegg til interaksjoner basert på *HGNC ID*, interaksjoner der kun en eller ingen av interaksjons protein/genene er *HGNC ID*. Disse vil bli fjernet under forbehandlingen. Se kapittel 5.2.

4.2.2.5 Hente ut data

Det som er viktig i denne fasen er å få hentet ut data fra de forskjellige databasene og gjøre dem klare for lesing. Ideelt sett skal dette foregå automatisk med fastlagte tidsintervaller. Dette blir ikke gjort i prototypen som er utviklet i forbindelse med denne oppgaven. Grunnene til dette er for det første at dette ikke har vært et hovedfokus og dermed ikke ville bidra til målet med oppgaven. For det andre vil det ikke være hensiktsmessig å bruke en fullstendig utgave av annoteringsdatabasene for å teste relasjonsdatabasen. Dette vil bli uoversiktlig og mulighetene for å se hvorvidt relasjonsdatabasen fungert optimalt vil bli vanskeligere. Dessuten vil det å fylle relasjonsdatabasen med fullstendige utgaver av alle databasene ta svært lang tid og testing vil ikke gi noen gode testresultater.

UniProtKB/Swiss-Prot finnes blant annet i XML-format og flat fil og kan lastes ned som gzip-fil. flat fil inneholder all informasjonen som er nyttig i denne sammenheng, men er strukturert på en slik måte at det vil være ressurskrevende å hente ut de

nødvendige data. XML-format blir derfor valgt fordi det vil være enklest å hente ut data fra denne fila ved hjelp av XML-taggene.

HGNC-databasen kan lastes ned som html eller tabulatordelt tekstfil. Siden det er forholdsvis enkelt å hente ut tabulatordelt tekst, og denne fila er mer strukturert enn html fila, blir denne filtypen valgt. Dette er ikke en zip-fil, kun en flat fil.

BioGrid og IntAct tilbyr begge tabulator-delt tekstfil, som kan lastes ned som zip-fil. Siden begge kommer i samme format og strukturen i fila er tilnærmet lik, vil det være mest egnet i denne sammenheng, med tanke på metoden som skal brukes for å hente ut data fra filene.

Siden de fleste av databasene ikke kan lastes ned i samme format og heller ikke har samme struktur, vil det måtte finnes en metode for hver database for å hente ut data. Dette vil fungere så lenge det bare fire stykker, men hvis det skal tilføres flere database, kan det bli mange metoder. Her er det viktig så langt det er mulig å bruke samme format på databasene, siden det vil gjøre uthenting av data mindre komplisert.

4.2.3 Forbehandling av data

Det er viktig at oppføringene i ordboka samsvarer med protein-/genforekomstene i abstraktene, når det gjelder tegn, format og lignende, slik at de enkelt kan bli gjenfunnet. Ordboka må derfor også forbehandles i samsvar med abstraktene. I kapittel 2.1 ble de forskjellige mulige stegene i forbehandling av biomedisinsk tekst omtalt. Ikke alle disse er aktuelle i dette tilfelle og noen ekstra steg som ikke direkte har med tekst behandling må tilføres

4.2.3.1 Vanlige forbehandlingssteg

Apostrofer

En del oppføringer både i UniProtKB/Swiss-Prot og HGNC inneholder apostrofer. Fjerning av disse kan føre til at protein-/gennavnet blir endret til et ugjenkjennelig protein-/gennavn eller til et helt annet protein-/gennavn. For eksempel en fjerning av apostrofer i protein-/gennavnet *`3' 5' exoribonuclease (RRP4 )`,* vil føre til *`3 5 exoribonuclease (RRP4 )`* eller *`35 exoribonuclease (RRP4 )`.* Hvis man fjerner apostrofer både i relasjonsdatabasen og i abstraktene, vil fortsatt protein-/gennavnen kunne bli identifisert, men de vil være til liten nytte for en bruker, som kan

bli forvirret over resultatet som kommer ut. Apostrofer vil derfor i denne sammenheng ikke fjernes.

Bindestrek

Fjerning av bindestreker er ikke så kritisk som fjerning av apostrofer. Men hvis man fjerner dem er det vanskelig å vite om de skal erstattes med mellomrom eller ikke. Hvis ikke mellomrommet fjernes vil enkelte protein-/gennavn bli ugjenkjennelige. For eksempel kan protein-/gennavnet `*angio-associated, migratory cell protein*` bli til `*angio associatedmigratory cell protein*`. For andre protein-/gennavn som `*BCL2 binding component 3*` og `*bcl-2-binding component 3*`, som begge finnes i UniProtKB/Swiss-Prot vil man fortsatt kunne ende opp med to forskjellige skrivemåter for et protein/-gennavn hvis man erstatter bindestrek med mellomrom og man har ingen garanti for at det er det tredje alternativet med mellomrom som finnes i abstraktet(`*BCL 2 binding component 3*`). Istedenfor å fjerne bindestrekene, vil det dermed sannsynligvis være mer gunstig å bruke tilnærmet likhet for å kunne identifisere protein-/gennavn. Dette vil føre til at relasjonsdatabasen inneholder flere versjoner av samme protein. For å kunne finne ut hvilken av tilnærmet like oppføringer som er riktig kreves det at eksperter på området manuelt plukker ut hvilken riktig versjon av navnet, noe som er utenfor rekkevidden av denne oppgaven.

Flere formater

Svært mange protein-/genoppføringer fra annoteringsdatabasene inneholder tall og det kan finnes skriveforskjeller mellom oppføringene i relasjonsdatabasen og abstraktene, samt innad i relasjonsdatabasen. Dette dreier seg likevel mer om tegn som for eksempel apostrof og bindestrek, enn om standard tallformat og er derfor ikke aktuelt når det gjelder protein-/gennavnene.

Setnings grense påvisning

Når det gjelder setnings-påvisningstegn, som utropstegn og spørsmålstegn, forkomme ikke disse i annoteringsdatabasene. Punktum derimot, forekommer i protein-/gennavn på lik linje med apostrof og bindestrek. Det vil ikke være noen grunn til å fjerne punktum fra protein-/gennavnene, da dette vil by på samme problemer som ved fjerning av apostrof og bindestrek. Punktum vil i denne

sammenheng uansett ikke være noen god måte å finne setningsgrenser på, siden mange protein-/gennavn inneholder punktum.

Andre tegn

Andre tegn som også forekommer i protein-/gennavn er komma, kolon og semikolon. Ingen av disse bør fjernes av samme grunner som for apostrof, bindestrek og punktum.

Morfologisk analyse

Protein-/gennavn kan få flertallsendelser, men ikke bøyninger. Dette gjelder i tilfelle i abstraktene og vil ikke finnes i protein-/gendatabasene. Flertallsendelser i abstraktene må derfor analyseres og normaliseres. Dette vil gjøre identifiseringen av navn og interaksjoner enklere.

Stoppord

Ved å bruke en stoppordliste beregnet på vanlig tekst, kan man risikerer å fjerne deler av protein-/gennavn. Som Moon og Singh [19] påpeker kan en spesial-tilpasset stoppord liste være hensiktsmessige for informasjonsgjenfinning i dette domenet. Dette er utenfor området til denne oppgaven.

Det er på grunnlag av dette ikke aktuelt å foreta en fjerning av stoppord i dataene som skal inn i relasjonsdatabasen, fordi dette kan føre til at protein-/gennavn og -synonymer mister sin opprinnelige mening og blir ugjenkjennelige. Dette forutsetter også at stoppord ikke fjernes i abstraktene.

4.2.3.2 Annen forbehandling

Det finnes forbehandlings steg som ikke direkte går under kategoriene leksikalsk, syntaktisk eller semantisk nivå, men som kan være aktuelle i denne sammenheng.

Lange oppføringer

En del oppføringer i protein-/gendatabasene består av veldig mange ord. Disse er ikke hensiktsmessige å ha med fordi de sjelden vil finnes i abstrakter. Alle navn må derfor sjekkes for antall ord og de oppføringene som inneholder over 10 ord blir ikke lagt til i interaksjonsdatabasen.

Oppføringer uten mening

Dette gjelder oppføringer som ikke vil bidra til noe, bare gjøre ordboka dårlig. I IntAct og BioGrid ble dette gjort manuelt, men det kan også gjøres, ved å sjekke hver oppføring og se om den starter med hgnc eller uniprotkb. Når det gjelder UniProtKB/Swiss-Prot og HGNC har det ikke blitt foretatt noen slik analyse, fordi det ikke forekommer noen liste over hvilke oppføringer som, i biomedisinsk sammenheng, er uten mening.

Oppføringer som ikke er unike

Samme oppføring kan forekomme i UniProtKB/Swiss-Prot og HGNC, men disse skal begge legges inn i relasjonsdatabasen som en oppføring med et eller flere symbol/id fra hver annoteringsdatabase. Like oppføringer innad i en database kan som nevnt også oppstå. Disse legges til med alle symbol/id-er i relasjons-databasen. For interaksjonsdatabasene blir dobbeltoppføringer eliminert når interaksjonene legges inn i databasen, siden det sjekkes om oppføringen ligger der fra før.

Fjerne navn som starter med vanlige utrykk

Vanlige uttrykk ville vært et problem hvis indeksen kun skulle bestått av enkeltord. I dette tilfelle skal den bestå av hele protein-/gennavn og et enkeltord i seg selv vil ikke gi treff, med mindre det er en protein-/genoppføring i databasen.

Gjøre store bokstaver til små

For å kunne gjøre søket uavhengig av store og små bokstaver bør både relasjonsdatabasen og abstraktene gjøres om til tekst kun bestående av små bokstaver. Dette blir gjort i prototypen, på protein-/gennavn og –synonymer, før de legges i relasjonsdatabasen.

4.2.4 Metoder for lagring av ordbok

Det finnes flere måter å lagre en ordbok på. De metodene som er aktuelle og vurderes er flat fil, relasjonsdatabase og indeks. En kombinasjon av disse kan også være mulig. I det følgende blir det diskutert hvilken som egner seg best i denne sammenhengen.

4.2.4.1 Flat fil

En ordbok i form av en flat fil vil ta liten plass og være forholdsvis rask å søke i. Data i en flat fil kan lagres på flere måter. De kan være delt ved tabulator, komma eller vertikalstrek. En flat fil kan også inneholde to-bokstav koder som identifiserer de forskjellige feltene i en oppføring.

Men selv om en flat fil er enkel, tar liten plass og er lett å finne fram i vil den ikke være spesielt gunstig til søking. Hovedsakelig fordi man må søke gjennom mye urelevant informasjon For eksempel for å identifisere protein-/genforekomster vil man også måtte søke gjennom synonymer og muligens også interaksjoner, hvis disse også legges inn i samme fil.

Problemer kan oppstå på flere områder. Oppdateringer fra annoterings og -interaksjonsdatabasene og relasjoner er hovedutfordringene. Det kan bli vanskelig å få oppdatert en slik fil uten å lage den på nytt hver gang man skal foreta en oppdatering fra databasene som brukes. Når det gjelder relasjoner vil man ikke kunne finne oppføringer som for eksempel har samme synonym på en like enkel måte som i en relasjonsdatabase. Dessuten vil det bli mye dobbeltlagring av data. Når det gjelder identifisering av interaksjoner, har man behov for en kobling fra interaksjonsdatabasene til de tilhørende protein-/genoppføringen, fordi databasene med relasjoner kun innholder id/symboler og ikke protein-/gennavn. Dette vil ikke la seg gjøre på noen tilfredsstillende måte, ved bruk av flat fil.

Hvis det på noen måte skal være gunstig å bruke flat fil må den legges i en indeks, men selv om det letter søkingen, vil det fortsatt oppstå problemer når det gjelder interaksjoner, synonymer og oppdatering.

4.2.4.2 Relasjonsdatabase

En felles relasjonsdatabase som inneholder protein-/gennavn, synonymer, interaksjoner og annen nyttig informasjon og som oppdateres med jevne mellomrom vil kunne bidra til effektivitet ved identifisering av protein og protein interaksjon. Det gir dessuten rom for relasjoner som kan gi brukeren nyttig informasjon i tillegge til protein-/gennavn, synonymer og interaksjoner.

Ved bruk av relasjonsdatabase kan man enkelt hente ut data og relasjoner, som for eksempel proteiner for samme synonym. Det som er ulempen med bruk av

relasjonsdatabase er at søking direkte i den ved identifisering av proteiner vil bli ressurs og tidkrevende, spesielt siden det i dette tilfelle er snakk om forholdsvis store mengder data.

Det aller største problemet ved å bruke en relasjonsdatabase er oppdatering, men også det å legge inn data første gang. Det er vanskelig å oppdatere en database uten at det blir svært ressurskrevende. Problemer som kan oppstå er blant annet at databasen er i bruk, altså det søkes i den, samtidig som den oppdateres. Dette vil ikke være gunstig og kan føre til at brukeren ikke får ut fullstendig informasjon. Det er også problemer knyttet til hvordan selve oppdateringen skal foregå. Skal man fylle hele databasen med data på nytt hver gang den skal oppdateres, eller går det an på en enkel måte å legge til og fjerne aktuelle oppføringer.

En mulighet som kan løse mange av problemene nevnt ovenfor er en kombinasjon av relasjonsdatabase og indeks, der relasjonsdatabasen indekseres før søking. Dette vil føre til raskere søk og kan også løse enkelte av problemene som oppstår ved oppdatering av relasjonsdatabasen. Dette kan i praksis foregå på flere måter og føre til at databasen ikke blir involvert i selvet identifiseringen av protein-/gen-forekomster.

For at relasjonsdatabasen skal kunne bli brukt effektivt til å identifisere protein-/gen-forekomster i abstrakter, må lagringen av informasjon oppfylle visse kriterier. For det første må man i størst mulig grad unngå dobbeltlagring. Dette er en generell regel for konstruksjon av relasjonsdatabaser og er viktig for å kunne gjøre databasen minst mulig og best mulig strukturert. Det blir ekstra viktig når det er snakk om mye data som skal inn i databasen. Mye dobbeltlagring vil da kunne få store konsekvenser for størrelsen og tiden det tar å søke og hente ut informasjon fra databasen.

Fokuset i relasjonsdatabasen bør være på protein-/gennavn, siden det er disse som skal identifiseres i abstrakter. Dette er også viktig med tanke på identifisering av interaksjoner, siden disse kun er oppgitt med symboler og id fra UniProtKB/Swiss-Prot og HGNC.

Relasjonsdatabasen bør også være mest mulig generell, slik at den ikke kun kan brukes for å lagre de databasene som er brukt som testdatabaser her. Dette er

viktig fordi en spesifikk relasjonsdatabase kun tilbruk for utvalgte databaser ikke vil være til noen generell nytte.

4.2.4.3 Indeksering

En indeks kan i dette tilfelle bygges på to måter, fra flat fil eller en relasjons-database. Uavhengig om flat fil eller relasjonsdatabase blir valg til å lagre data fra de forskjellige annoterings- og interaksjonsdatabasene, må dataene indekseres. Dette fordi et søk direkte i en relasjonsdatabase eller en flat fil vil være for tid og ressurskrevende.

Indeksering har tradisjonelt blitt brukt i forbindelse med søk etter dokumenter, web sider etc. Da er det dokumentet som blir indeksert. I dette tilfellet er det en allerede eksisterende samling indekstermer som skal indekseres. Og dokumentet som skal brukes til å finne termene. Dette blir litt omvendt av tradisjonell indeksering.Det finnes flere muligheter når det gjelder hvordan man skal bygge en indeks. Disse ble presentert i kapittel 2.5.

Inverterte filer

Det er to ting som taler for at inverterte filer er en indekseringsmåte egnet for tilnærmelsen i denne oppgaven. For det føreste er formålet med inverterte filer å gjøre gjenfinning mer effektiv og redusere størrelsen på plassen som trengs for å lagre forekomstene av alle indekstermene. Dette vil være svært nyttig ved bruk av en relasjonsdatabase, som kan inneholde svært mye data. Dermed vil identifisering av protein-/gennavn og -synonymer kunne foregå raskere enn ved direkte søk mot databasen, noe som er hovedargumentet for å indeksere relasjonsdatabasen. Et annet argument for å bruke inverterte filer er at de er effektive fordi man ikke trenger å gå gjennom hele relasjonsdatabasen for hvert ord i en spørring. Et protein-/gennavn og -synonym består ofte av flere ord. Hvis man skulle foreta søk direkte mot relasjonsdatabasen ville det måtte søkes på hvert ord i spørringen, for å finne oppføringer i relasjonsdatabasen, som inneholdt det gitte ordet. Ved bruk av indeks vil man slippe å gå gjennom hele databasen. [5]

Lucene

Som nevnt i kapittel 2.5.1 vil Lucene være egnet for indeksering i denne sammenheng. Grunnen til dette er at Lucene tilbyr et fulltekst søkemotor bibliotek, skrevet i Java. Dette kan tilpasses den samlingen man ønsker å søke i og gjør

indeksering og søking enkel. I kapittel 2.5.1.1 blir de forskjellige feltene som kan brukes i en indeks bygget ved hjelp av Lucene presentert. Det er ikke alle de tilgjengelige feltene som er aktuelle i denne sammenheng. Det som vil ligge i indeksen er protein-/gennavn, -synonym og en merkelapp for hvorvidt det er et protein-/gennavn eller et -synonym. Siden både protein-/gennavn og -synonym er primærnøkkel i tabellen vil det ikke være nødvendig med noen kobling mot databasen utover dette. Hvis et annet felt enn det som skulle ligge i indeksen var primærnøkkel, måtte også dette bli lagt til som en kobling mot oppføringen i relasjonsdatabasen.

Protein-/gennavn og -synonymer vil ikke bli forbehandlet utover fjerning av lange oppføringer og omgjort til småbokstaver. Feltet *Keyword* er derfor mest egnet for å lagre protein-/gennavn og -synonymer i indeksen, fordi det ikke blir forbehandlet men indeksert og lagret i indeksen i sin helhet. Problemer med å legge hele strenger, i form av protein-/gennavn, i indeksen kan oppstå hvis et protein-/gennavn staves annerledes i et abstrakt enn i indeksen. Dette løses ved tilnærmet treff som enkelt kan implementeres i Lucene. Et annet alternativ for å lagre protein-/gennavn i indeksen er feltet *Text*, men dette feltet blir analysert, noe som vil si at protein-/gennavnene ikke kan legges der i sin helhet og dermed vil bli splittet i enkeltord.

Feltet *UnStored* er ikke et aktuelt felt i denne sammenheng fordi det ikke finnes store mengder tekst, som for eksempel dokumenter, i databasen. Det feltet som er aktuelt i tillegg til *Keyword* er derfor *UnIndexed* som i dette tilfelle vil inneholde hvilken tabell oppføringen kommer fra og dermed identifisere om elementet er et protein-/gennavn eller et synonym. Feltet *UnIndexed* er egnet til dette fordi det verken blir forbehandlet eller indeksert, men blir likevel lagret i indeksen. Dermed kan det ikke søkes på, men kan hentes ut sammen med de identifiserte protein-/gennavnene og -synonymene for å kunne finne ut hvorvidt det er et protein-/gennavn eller -synonym.

En indeks kan i følge Gospodnetic og Hatcher [13] lagres på disk eller være i minnet så lenge applikasjonen kjører. En indeks lagret i minnet vil være raskere å søke i, men for en indeks med store mengder data vil det kreves mye minne for å

kunne gjøre dette. Begge typene indeks vil i dette tilfelle bli bygd hver gang prototypen starter.

Fordelen med en indeks lagret i minnet er at den i tillegg til å være raskere ved søk også er raskere å opprette. I dette tilfelle blir en indeks holdt i minnet brukt. Grunnen til dette er at mengden data som blir lagt i relasjonsdatabasen ved testing ikke er spesielt stor. En indeks som kun ligger i minnet vil derfor være egnet for rask indeksering og raske søk.

Et annet argument for å opprette indeksen hver gang prototypen starter er at prototypen ikke tar for seg oppdatering, endring av indeksen, samt at indeksen bli liggende lokalt. Dette vil si at for at det skal være gunstig å lagre en indeks permanent på disk, må applikasjonen være web-basert, slik at databasen og indeksen ligger på en sentralisert server. Dette er en mulig videreutvikling av prototypen og vil bli diskutert nærmere i kapittel 7.

Suffiks-tabeller og -trær

Som nevnt i kapittel 2.5 er suffiks-tabeller og -trær ressurskrevende å bygge, ikke spesielt egnet for store tekster og er ikke passende for tilnærmede tekst søk. Dette er to argumenter mot at suffiks-tabeller/trær er aktuelle i denne sammenheng. For det første kan relasjonsdatabasen bli svært stor, ettersom hvor mange annoterings- og interaksjonsdatabaser man kombinerer. Dessuten kan det bli aktuelt med tilnærmede tekst søk for eksempel fordi protein-/gennavn kan være skrevet forskjellige i et abstrakt og i relasjonsdatabasen.

Signatur-filer

En fordel med signatur-filer, er at de passer for veldig mye tekst, noe som taler for at det kan være aktuelt å bruke sammen med en relasjonsdatabase. Samtidig er det en bakdel at det er mulig å få treff selv om ordet forespurt ikke er i blokken for den passende bit-masken.

Ved bruk av en hash-tabell vil man for eksempel finne et ord i abstraktet som gir treff på et ord i rota av signatur-fila. Så går man videre til neste ord i abstraktet og ser om dette gir treff med noen av greinene. Hvis det ikke likhet med noen av greinene til rota, vil rota skrives ut som et treff. Dette kan da være kun en del av et

protein-/gennavn og ikke ha noen mening i seg selv. Noe som vil føre til falske treff og lavere presisjon.

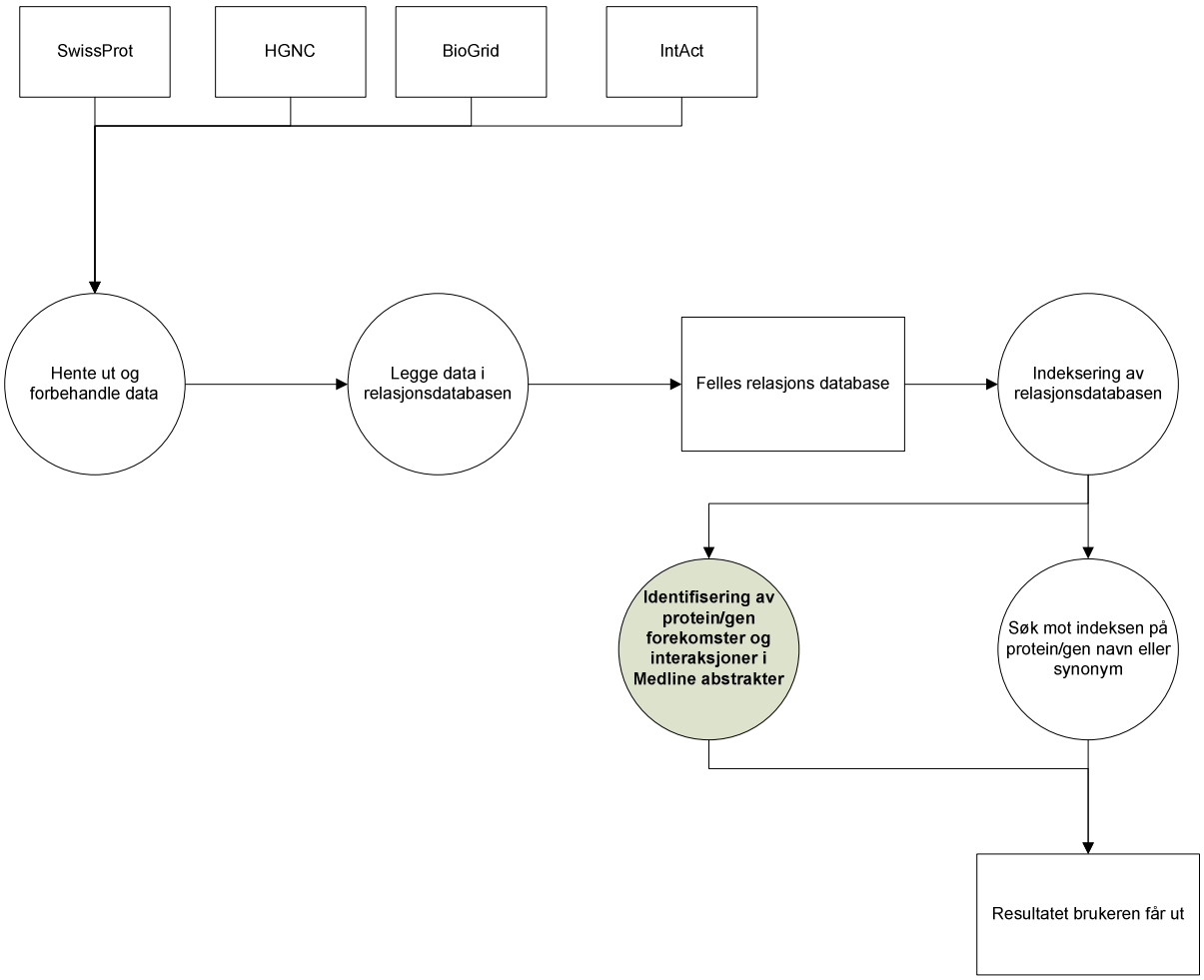
4.2.5 Resultat

Identifisering av protein-/genforekomster og -interaksjoner gir i seg selv nødvendigvis ikke brukeren god nok informasjon. Databasen må derfor være konstruert på en slik måte at den kan gi brukeren mer informasjon om resultatet. Dette er en av hovedgrunnene, i tillegg til identifisering av interaksjoner, for å lagre symbol/id i relasjonsdatabasen sammen med protein-/gennavnet og -synonymene. Det er også hovedargumentet for å lagre protein-/gennavn og -synonymer i hver sin tabell. En slik løsning vil gjøre at brukeren i tillegg til å få opplyst hvilke protein-/gennavn eller -synonym forekomster som er identifisert, få informasjon om symbol/id, interaksjoner og kryssreferanser. Symbol/id kan igjen brukes til å hente ut mer informasjon fra annoteringsdatabasene, enten automatisk, som en utvidelse av prototypen, eller manuelt ved at bruker selv søker i annoteringsdatabasene. Dermed kan brukeren få ut informasjon som kan hjelpe brukeren videre.

5 Implementasjon

Implementasjonen består av tre hoveddeler. Først blir informasjonen som skal brukes hentet ut fra filene lastet ned fra annoterings- og interaksjonsdatabasen. Informasjonen blir så lagt i en relasjonsdatabase og deretter indeksert. I indeksen kan man kan søke på protein-/gennavn og -synonymer. Som et resultat av søk mot indeksen får man ut protein-/gennavn og -synonymer med tilleggsinformasjon som hentes ut fra relasjonsdatabasen.

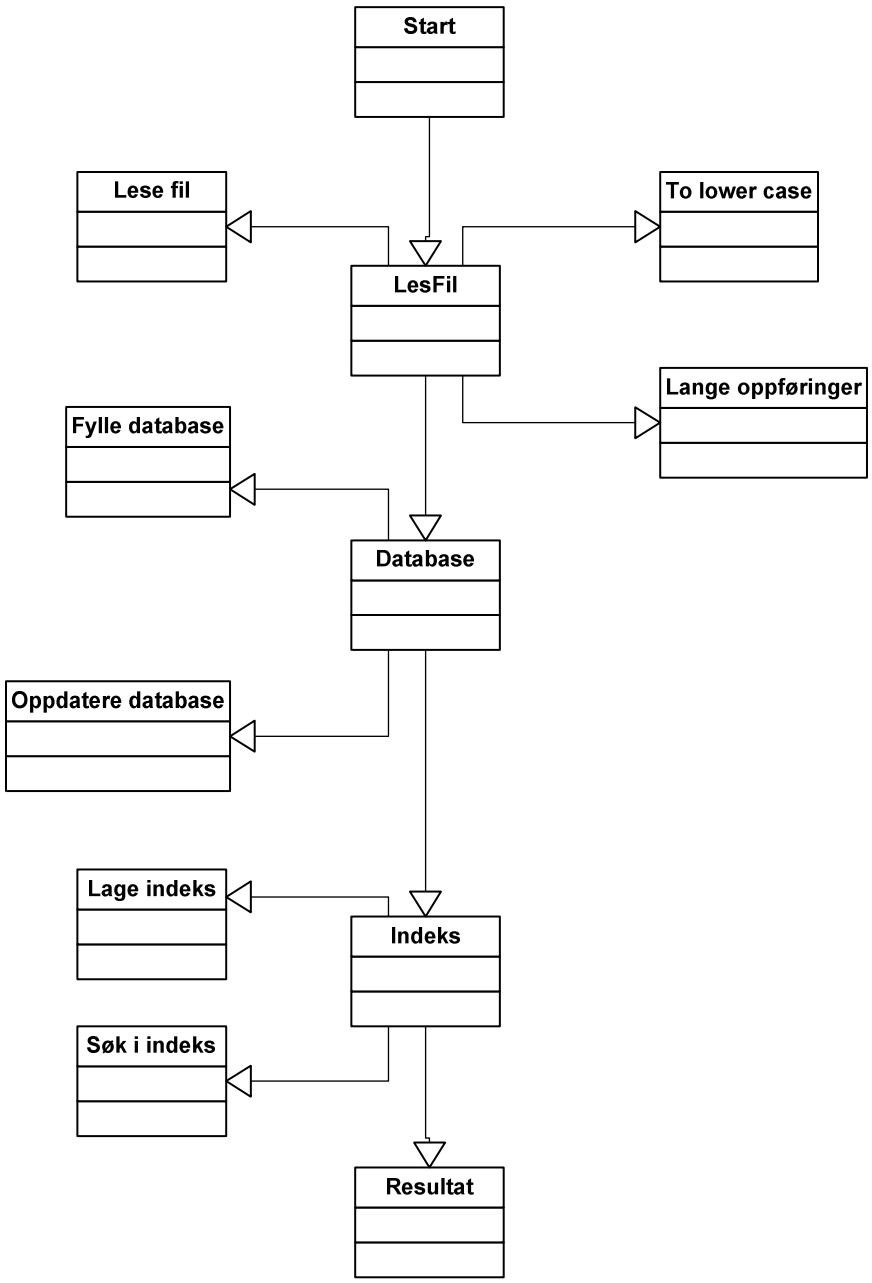
Implementasjonen underbygger forslaget for hvordan man på en gunstig måte kan kombinere flere annoterings- og interaksjonsdatabaser i en relasjonsdatabase.



Figur 2 Modell av systemet

Som man kan se av Figur 2 blir først data hentet ut fra databasene, før de blir forbehandlet og lagt i relasjonsdatabasen. En indeks lages deretter av dataene i

relasjonsdatabasen. I denne prototypen kan man søke i indeksen og få ut et resultat som viser treff og tilleggsinformasjon fra relasjonsdatabasen. Ideelt sett skal indeksen brukes til direkte å identifisere protein-/gennavn og -interaksjoner i artikkelsammendrag. Dette vises som en grå sirkel, men er utenfor omfanget av denne oppgaven.



Figur 3 Klassediagram

Figur 3 viser klassediagrammet for prototypen. Det klassediagrammet imidlertid ikke viser er at det finnes forskjellige klasser for uthenting og lagring av data fra de forskjellige annoterings- og interaksjonsdatabasene. Oppbyggingen er likevel

prinsipielt lik og det er derfor valgt å gi en oversikt over strukturen i prototypen i steden for hver enkelt klasse.

Grunnen til at det er utviklet forskjellige klasser for enkelte av annoterings- og relasjonsdatabasene, er at det å hente data ut fra dem og legge disse i relasjonsdatabasen, forgår forskjellig fra database til database. De eneste databasene som kunne bruke samme klasser var interaksjonsdatabasene fordi disser er like i filformat og struktur.

5.1 Hente ut data fra filene

Testdata er lastet ned fra de aktuelle databasene. UniProtKB/Swiss-Prot ble lastet ned i XML-format. De andre databasene ble lastet ned i tabulator-delte filer. Grunnen til dette er at det kun var UniProtKB/Swiss-Prot som kunne lastes ned i XML. XML kan enkelt analyseres og man kan hente ut kun de feltene som er aktuelle. For de andre databasene ble all informasjon i filene først hentet ut for så å plukke ut de aktuelle feltene.

For å kunne test databasen, vil det være lite gunstig å bruke hele UniProtKB/Swiss-Prot-databasen, siden den er svært stor og det vil ta enormt lang tid å få lagt den inn i relasjonsdatabasen. Dessuten vil det bli svært uoversiktlig med så mye data i relasjonsdatabasen, noe som vil føre til at det er vanskelig å se om implement-asjonen og relasjonsdatabasen fungerer optimalt. Et testsett fra UniProtKB/Swiss-Prot er derfor hentet ut, der det både finnes oppføringer med like protein-/gennavn, protein-/gennavn som er synonymer for andre protein-/gennavn og-synonymer som tilhører flere protein-/gennavn.(Kapittel 6.1)

HGNC er bare noen få megabyte i sin helhet, lastet ned i tabulator-delt tekstfil. For å kunne sjekke om den fungerer opp mot UniProtKB/Swiss-Prot er det også fra den hentet ut et testsett. Dette består av protein-/gennavn som har referanse til UniProtKB/Swiss-Prot-oppføringene i testsettet. En del av oppføringene i HGNC har ingen referanse til UniProtKB/Swiss-Prot og for å teste prototypen, og relasjonsdatabasen er også en del slike oppføringer tatt med i testsettet.(Kapittel 6.1)

Når det gjelder interaksjonsdatabasene er det også fra disse hentet ut testsett. Oppføringer som ikke har UniProtKB/Swiss-Prot-symbol eller HGNC-id fjernet.

Når filene er lastet ned må data hentes ut. Det er individuelt for hver database, hvordan dette skal foregå, siden de hentes ut i forskjellige filformat og filene har forskjellig intern struktur.

XML-fila fra UniProtKB/Swiss-Prot-databasen analyseres. Det vil si at kun de dataene man ønsker hentes ut fra fila ved hjelp av XML-taggene. I dette tilfelle hentes symbol, synonymer ut fra UniProtKB/Swiss-Prot-databasen. Synonymene for hvert protein/gen legges i en liste.

For hver oppføring som hentes ut av UniProtKB/Swiss-Prot XML-fila, kalles klassen som legger oppføringen i databasen opp. Dette kan føre til at prototypen får litt lengre kjøretid en det som er optimalt, men samtidig unngår man at store mengder data må lagres i minnet samtidig og man kan legge forholdsvis store filer i databasen, uten å ha et enormt stort minne.

HGNC-databasen er lastet ned i tabulator-delt tekstfil. For å hente ut data fra denne må den skilles på tabulatorer. Et problem som oppstod her, er at enkelte av oppføringene har tomme felter. For eksempel var det en del oppføringer som ikke hadde synonymer. Dermed ble det vanskeligere å bruke tabulator, fordi den bare skilte mellom tekst og ikke tekst. Løsningen på dette var å forbehandle fila ved å sette inn ’null’ i alle de tomme feltene. Dette ble gjort manuelt.

Den store forskjellen mellom disse to databasene var hvordan oppføringene ble hentet ut fra fila. Ellers ble også HGNC-synonymene puttet i en liste for hver protein-/genoppføring som i UniProtKB/Swiss-Prot.

Filene fra interaksjonsdatabasene ble som nevnt forbehandlet ved å fjerne interaksjoner som ikke var på formen HGNC-id eller UniProtKB/Swiss-Prot-symbol. Siden begge databasene kunne lastes ned i tabulator-delt tekstfil og har forholdsvis lik struktur, kunne samme metode brukes for å hente ut data.

5.2 Forbehandle data

Ut fra vurderingene foretatt i kapittel 4.2.3 er det lite forbehandling som skal foretas på dataene som hentes ut fra annoterings- og interakasjonsdatabasen. Når dataene fra annoteringsdatabasene har blitt hentet ut fra filene blir det sjekket om

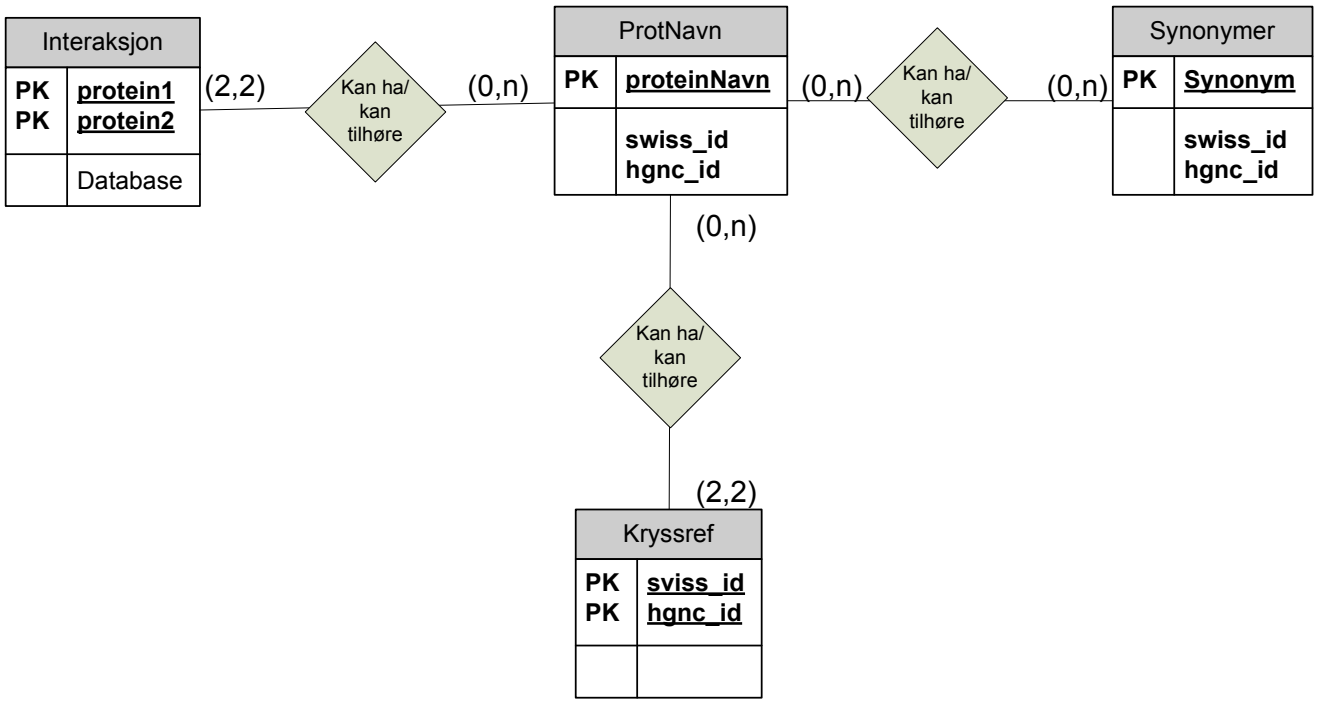
protein-/gennavn eller -synonym er over ti ord lange. Hvis dette er tilfelle, blir oppføringen forkastet. Hvis oppføringen derimot inneholder ti eller færre ord, blir protein-/gennavnet eller -synonymet gjort om til små bokstaver.

Når det gjelder interaksjonsdatabasen blir disse som tidligere nevnt forbehandlet ved å fjerne oppføringer der en eller begge av interaksjonene ikke starter med uniprotkb eller hgnc, for henholdsvis IntAct og BioGrid. Når data er hentet ut av filene blir disse merkelappen fjernet fra oppføringene, slik at man kun står igjen med symbol eller id.

5.3 Relasjonsdatabase

5.3.1 Struktur

Etter forbehandling legges dataene i relasjonsdatabasen. Relasjonsdatabasen inneholder, som man ser av Figur 4, fire tabeller.



Figur 4 ER-modell av relasjonsdatabasen

ProtNavn-tabellen er hovedtabellen i databasen og er den tabellen som kobler de andre tabellene sammen. *ProtNavn* inneholder tre felter protein-/gen-navn(proteinNavn), UniProtKB/Swiss-Prot-symbol(swiss_id) og HGNC-id(hgnc_id).

Primærnøkkelen er *proteinNavn*, fordi dette er unikt og på den måten unngår man også at protein-/gennavn blir lagt til flere ganger i databasen, samt at man får dobbeltoppføringer. *Swiss_id* og *hgnc_id* feltene kan inneholde flere symbol/id fra hver annoteringsdatabase. Dette fordi et protein-/gennavn kan finnes i flere oppføringer. En oppføring i tabellen *ProtNavn* kan ha ingen eller mange synonymer, kryssreferanser og interaksjoner med andre protein/gen.

Tabellen *Synonymer* har samme struktur som *ProtNavn*. Den består av et protein-/gennavn og tilhørende UniProtKB/Swiss-Prot-symboler og HGNC-id-er. Et symbol i *Synonymer*-tabellen kan tilhøre mange oppføringer i *ProtNavn*.

Både protein-/genoppføringen i *ProtNavn*-tabellen og mange av synonymene i *Synonymer,* er protein-/gennavn. Det ville derfor spille liten rolle for identifiseringen av protein-/genforekomster om de lå i en tabell. Men veldig mange synonymer er ikke selvstendige oppføringer i noen av databasene og det vil derfor være nyttig å skille mellom dem. På den måten kan brukeren motta mer utfyllende informasjon om hva slags type oppføring som er identifisert i abstraktene.

Som nevnt tidligere er det slik at en oppføringe i HGNC kan ha en referanse til UniProtKB/Swiss-Prot, men det betyr likevel ofte ikke at protein-/gennavnene er identiske. Av den grunn må en relasjon mellom oppføringene likevel trekkes for å vise at de er samme protein/gen til tross for skriveforskjeller. Dette gjøres ved hjelp av tabellen *Kryssref*. Her legges kryssreferansen mellom de forskjellige protein-/genoppføringene i UniProtKB/Swiss-Prot og HGNC. Protein/gener uten kryssreferanse legges ikke i denne tabellen, fordi det ville tatt opp unødvendig mye plass og har ingen nytte. Disse kan heller legges til ved en seinere anledning hvis databaser med kryssreferanser til dem tilføres. Kryssreferansene består i dette tilfellet av en kombinasjon av et symbol fra UniProtKB/Swiss-Prot og id fra HGNC. Både HGNC-id og UniProtKB/Swiss-Prot-symbolet i *Kryssref*-tabellen må ha tilhørende oppføring i *PoteinNavn*-tabellen til for at kryssreferansen skal være til noen nytte, men det kan forekomme ved flere tilfeller at den bare tilhører et protein-/genoppføring. Dette kan skje hvis for eksempel en oppføring i HGNC kryssrefererer til en oppføring i UniProtKB/Swiss-Prot som ikke ligger i relasjonsdatabasen. Dette vil sannsynligvis ikke forekomme i stor grad hvis en fullstendig utgave av begge annoteringsdatabasene er lagt til i relasjonsdatabasen.

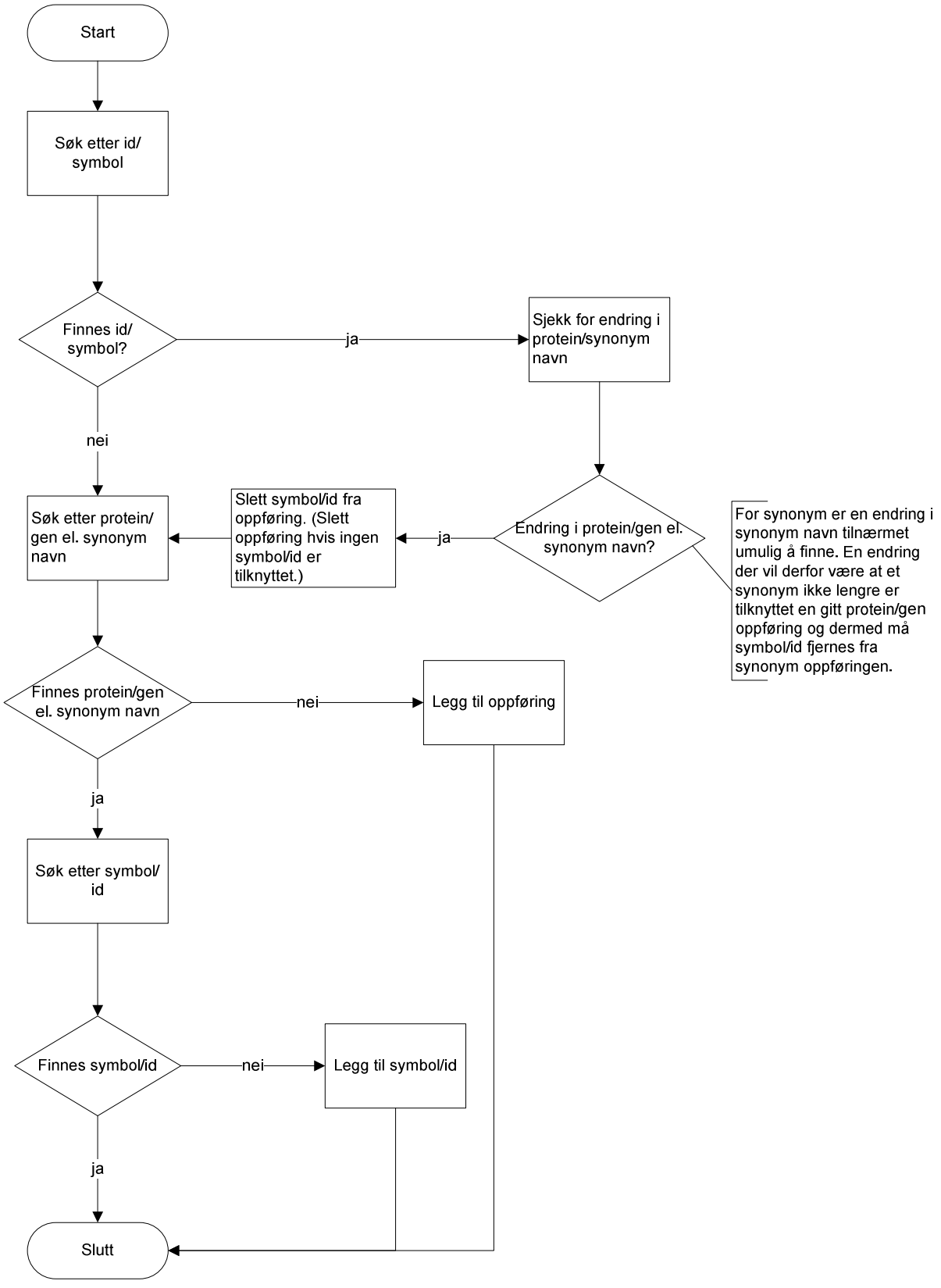
Selv om en kryssreferanse må tilhøre en protein-/genoppføring for å bli lagt i tabellen, kan man likevel risikere at oppføringer i *Kryssref*-tabellen referere til to protein-/genoppføringer som ikke finnes i relasjonsdatabasen. Dette kan komme av at oppføringer ha skiftet symbol eller blitt slått sammen, eventuelt at den har blitt slettet. Oppdatering av *Kryssref*-tabellen er derfor viktig.

5.3.2 Legge inn data

I dette tilfelle blir databasene UniProtKB/Swiss-Prot, HGNC, IntAct og BioGrid brukt til å teste databasen, men databasen er laget for at man skal kunne bruke en hvilken som helst protein-/gennavn eller -interaksjonsdatabase, med tilsvarende felter som de i relasjonsdatabasen.

Slik denne prototypen fungerer spiller det ingen rolle hvilken av annoterings-databasene som legges til i relasjonsdatabasen først. Grunnen til dette er at ingen av de to annoteringsdatabasene på noen måte samhandler med den andre når de legges til i relasjonsdatabasen.

Når det gjelder interaksjonsdatabasene, har heller ikke disse noe med hverandre å gjøre siden de inneholder referanser til hver sin annoteringsdatabase og kan legges til i relasjonsdatabasen i vilkårlig rekkefølge. De påvirker heller ikke annoterings-databasene eller blir påvirket av dem ved innfylling.



Figur 5 Metoden for å legge til og oppdatere protein-/gennavn og -synonymer.

Figur 5 viser et flytdiagram over metoden for å legge til og oppdatere protein-/gen-navn og -synonymer i relasjonsdatabasen.

5.3.2.1 UniProtKB/Swiss-Prot-oppføringer

Når en oppføring fra UniProtKB/Swiss-Prot er ferdig forbehandlet blir metoden som legger oppføringene i databasen kalt opp. Hvis det er første gang UniProtKB/Swiss-Prot-databasen blir lagt i relasjonsdatabasen, det vil si at det ikke ligger noen UniProtKB/Swiss-Prot-oppføringer der fra før, legges protein-/gennavn og -symbol til ved at det foretas et søk mot databasen for å se om protein-/gennavnet ligger der fra før. Hvis det finnes fra før, legges symbolet til i feltet *swiss_id* sammen med de eventuelle symbol/id-ene som ligger der fra før. Hvis ikke protein-/gennavnet finnes fra før, legges det til som en ny oppføring sammen med symbolet.

Den samme metoden brukes både første gang databasen fylles og til oppdatering. Dette vil ikke by på problemer fordi metoden først sjekker om symbolet ligger i databasen. Symbolet vil aldri gjøre det hvis databasen ikke er fylt fra før. Men databasen kan være fylt fra før og ikke inneholde et symbol, av den grunn at nye oppføringer kan være tilført UniProtKB/Swiss-Prot-databasen.

Tilføring av synonymer foregår på liknende måte ved første gangs fylling av databasen. Først blir det sjekket om synonymet ligger i *Synonymer*-tabellen fra før. Finnes det blir symbolet lagt til, hvis ikke blir en ny oppføring med synonym og symbol lagt til.

Hvis UniProtKB/Swiss-Prot er den første databasen som legges til i relasjons-databasen, vil det ikke være noe poeng å legge til kryssreferanser mot noen databaser. Dette kan gjøres fra databaser som tilføres etter at første database er lagt til. Hvis derimot, det allerede ligger en database i relasjonsdatabasen, kan det være aktuelt å legge til kryssreferanser. Det er ikke lagt inn noen kryssreferanser fra UniProtKB/Swiss-Prot i denne prototypen. Grunnen til dette er at det kun er behov for en kryssreferanse mellom UniProtKB/Swiss-Prot og HGNC og denne var enklere å hente ut fra HGNC.

Etter hvert som data legges inn UniProtKB/Swiss-Prot-databasen vil man få protein-/gennavn og -synonymer med flere symboler. Disse symbolene legges sammen i feltet *swiss_id*, kun skilt ved mellomrom.

5.3.2.2 HGNC-oppføringer

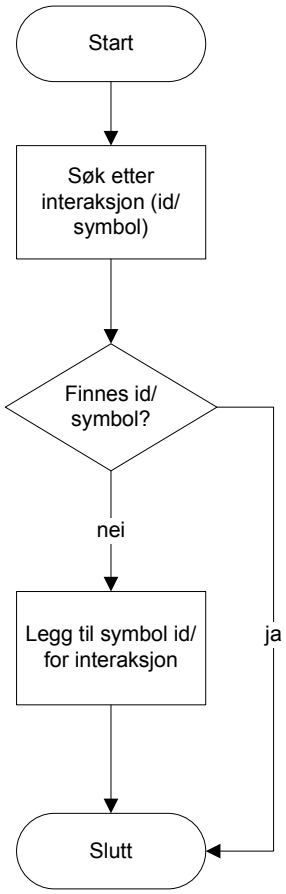
Å legge inn data fra HGNC foregår likt som med UniProtKB/Swiss-Prot-oppføringer, med unntak av at det også legges inn kryssreferanser til UniProtKB/Swiss-Prot.

Første gang HGNC-databasen legges i relasjonsdatabasen blir det sjekket om protein-/gennavnet ligger der fra før. Hvis det finnes, legges id til hvis den ikke finnes fra før, hvis ikke legges en ny oppføring til med protein-/gennavn og -id. Det samme gjelder for synonymer.

En del oppføringer i HGNC inneholder referanse til UniProtKB/Swiss-Prot-oppføringer. Disse legges i tabellen *Kryssref*. Metoden sjekker da først om hverken HGNC-id feltet eller referansefeltet med UniProtKB/Swiss-Prot-symbol er tomme. Hvis de ikke er det, blir det sjekket om kryssreferansen finnes i tabellen fra før. Er ikke dette tilfelle, legges kryssreferansen til. Id fra de oppføringene i HGNC som ikke inneholder kryssreferanse, blir da ikke lagt i *Kryssref*-tabellen.

5.3.2.3 Oppføringer fra IntAct og BioGrid

Oppføringene fra IntAct og BioGrid består kun av to felt med symboler/id som har en interaksjon. Disse oppføringene er helt avhengige av *ProtNavn*-tabellen for at de skal kunne identifiseres i abstrakter.



Figur 6 Flytdiagram over metoden for å legge til og oppdater interaksjoner i *Kryssref*-tabellen

Som Figur 6 viser legges interaksjonene inn i databasen ved at det først sjekkes om interaksjonene finnes i tabellen fra før. Hvis dette er tilfelle, blir ingen ting foretatt. Hvis interaksjonen ikke finnes legges den til. Tabellen består av to felt *protein1* og *protein2*, slik at hvert av symbol/id-ene blir lagt i hvert sitt felt.

En oppdatering av *Kryssref*-tabellen vil foregå på nøyaktig samme måte som første gangs innfylling i relasjonsdatabasen.

5.3.3 Oppdatering av databasen

Oppdatering av databasen foregår med samme metode som første gang data legges inn. Figur 5 viser hvordan.

Det som skjer ved en oppdatering av relasjonsdatabasen fra UniProtKB/Swiss-Prot eller HGNC er at det først blir sjekket om symbol/id allerede ligger i databasen. Hvis symbolet ikke finnes, kan det være fordi en ny oppføring har blitt opprettet i en av databasene. Da blir det samme som ved første gang man legger inn en oppføring utført. Det blir sjekket om protein-/gennavnet eller -synonymet allerede finnes i databasen. Er det tilfelle legges symbolet til. Hvis ikke, legges en ny oppføring til.

Hvis symbol/id finnes i databasen vil det bli foretatt en sjekk etter endringer i oppføringen og oppdatert hvis endringer er tilfelle. Dette foregår på litt forskjellige måter med protein-/gennavn og -synonymer.

For protein-/gennavn i *ProtNavn*-tabellen sjekkes det først om protein-/gennavnet i oppføringen med symbolet er det samme. Hvis det er det samme blir ingen endringer foretatt. Er derimot ikke dette tilfelle, blir symbolet slettet fra oppføringen og man søker etter protein-/gennavn tilsvarende det i oppføringen fra UniProtKB/Swiss-Prot. Finnes dette, legges symbolet til, hvis ikke legges en ny oppføringe til i databasen.

Når det gjelder oppdateringen av *Synonymer*-tabellen blir det også her sjekket om symbol/id finnes i relasjonsdatabasen fra før. Finnes det blir oppføringene med synonymet/id hentet ut, for i motsetning til *ProtNavn*-tabellen kan man her få mange treff. Alle treffene blir så gjennomgått for å sjekke om tilsvarende synonym blir funnet. Finnes det, blir ingen endringer foretatt. Hvis det ikke finnes derimot, blir det sjekket om synonymet finnes noen annen plass i tabellen. Er dette tilfelle tilføres symbol/id til synonymet, hvis ikke legges synonym med symbol/id til som en ny oppføring i tabellen.

I liket med *ProtNavn*-tabellen kan oppføringer bli endret eller slettet. Dette forgår på en litt annen måte enn med oppføringer i *ProtNavn*. Det er vanskelig å sjekke om et nytt synonym er et allerede eksisterende synonym, som har endring i navnet, eller om det er et nytt synonym. Det som kan gjøres er å sjekke om noen av

synonymene i relasjonsdatabasen ikke er tilknyttet protein-/gennavnet lengre og dermed slette symbol/id eller oppføringen.

Først blir alle oppføringene for et gitt symbol/id hentet ut fra relasjonsdatabasen. For hver av synonym navnene fra relasjonsdatabasen blir det sjekket om tilsvarende synonym finnes blant synonymene hentet ut, for oppføringen med samme symbol/id, fra annoteringsdatabasene.

Finnes et eller flere av synonymene fra relasjonsdatabasen ikke blant de som er hentet ut fra annoteringsdatabasen, blir enten symbol/id slettet fra oppføringen eller oppføringen blir slettet, hvis den ikke har flere symbol/id-er knyttet til seg.

Oppdatering av *Kryssref*-tabellen foregår på samme måte som ved første gangs innfylling. Fjerning av kryssreferanser som ikke lengre eksisterer, gjøres enklest ved å tømme tabellen og fylle den på nytt.

Som nevnt i kapittel 5.3.2.3 forgår en oppdatering av *Interaksjon*-tabellen også på samme måte som første gangs innfylling. Det sjekkes om interaksjonen ligger i tabellen fra før. Hvis den ikke gjør det, legges den til. Som med *Kryssref*-tabellen er den enkleste måte å fjerne interaksjoner, som ikke lengre eksisterer, å tømme tabellen og fylle den på nytt med jevne mellomrom.

5.4 Indeksering

For effektiv identifisering av protein-/genforekomster og -synonym i abstrakter blir relasjonsdatabasen indeksert. Dette foregår ved at protein-/gennavn som ligger i *ProtNavn* feltet i *ProtNavn*-tabellen blir hentet ut og lagt i en liste. Det samme gjelder for synonymer fra *Synonymer*-tabellen.

En indeks blir deretter opprettet i minnet og hver oppføring i listene fra relasjonsdatabasen blir lagt til, sammen med en merkelapp for hvilken tabell oppføringen er hentet fra, som et felt i indeksen.

5.5 Søk i indeksen

Poenget med å lage en indeks er å bruke denne til å identifisere protein-/gen-forekomster og -synonymer i abstrakter, samt å identifisere interaksjoner mellom disse.

Siden fokuset i denne oppgaven er på lagring av data i en felles relasjonsdatabase, er det kun implementert muligheter for søk på protein-/gennavn og -synonymer ved å skrive disse inn i et søkefelt.

Etter at relasjonsdatabasen er indeksert er det muligheter for å foreta søk i indeksen. Søk kan foregå på flere måter. Man kan søke på nøyaktig term, ved å skrive den inn i søkefeltet. Det er også muligheter for delvis treff ved bruk av såkalte jokertegnspørringer. Dette gjøres ved å bruke * for en eller flere tegn og ? for ingen eller et tegn. For eksempel spørringen *annexin** som vil føre til treff på både *annexin a11* og *annexin-b10-13*.

For hvert treff i indeksen blir det først sjekket om oppføringen er et protein-/gennavn eller et synonym. Dette gjøres ved å undersøke *UnIndexed* feltet som inneholder navnet på tabellen oppføringen er hentet fra.

Hvis resultat termen er et protein-/gennavn, blir alle symboler/id for navnet hentet ut fra *ProtNavn*-tabellen. Disse blir igjen brukt til å hente ut synonymer, interaksjoner og kryssreferanser for protein-/gennavnet.

Hvis resultat termen derimot er et synonym, blir alle symboler/id for protein-/gennavnet i *ProtNavn*-tabellen, som har det aktuelle synonymet knyttet til seg, hentet ut og gitt som resultat til brukeren.

6 Test

6.1 Testsett

6.1.1 Testsett for første gangs fylling av databasen

For å kunne teste ut om relasjonsdatabasen fungerer slik den skal, vil det være ugunstig å bruke de fullstendige utgavene av annoterings- og interaksjons-databasene. Et testsett er derfor hentet ut fra databasene. De vil føre til at relasjonsdatabasen kan testes for tidligere beskrevet utfordringer ved å kombinere flere annoterings- og interaksjonsdatabaser.

Et testsett for UniProtKB/Swiss-Prot er hentet ut, basert på følgende kriterier.

Testsettet må inneholde tilfeller av oppføringer:

- med samme protein-/gennavn.
- som også finnes i HGNC-databasen.
- som inneholder synonymer.
- med samme synonym.
- der en oppføring er et synonym for en annen oppføring og omvendt.
- som er tilnærmet like. Kun små forskjeller, som for eksempel med og uten bindestrek.

Ut fra disse kriteriene ble et testsett med 330 oppføringer fra UniProtKB/Swiss-Prot plukket ut. Av disse er 287 unike protein-/gennavn. Oppføringene har til sammen 843 synonymer der 718 er unike.

HGNC-databasen er i motsetning til UniProtKB/Swiss-Prot ikke så stor, men for å få testet relasjonsdatabasen hentes et testsett ut basert på samme kriterier som for UniProtKB/Swiss-Prot, samt at det både må finnes oppføringer med referanser til UniProtKB/Swiss-Prot-testsettet, referanser til UniProtKB/Swiss-Prot-oppføringer som ikke finnes i testsettet og oppføringer uten UniProtKB/Swiss-Prot-referanse.

Testsettet fra HGNC består av til sammen 300 oppføringer, hvorav 211 inneholder kryssreferanser i form av UniProtKB/Swiss-Prot-symboler som også finnes i UniProtKB/Swiss-Prot-testsettet. Det inneholder også 40 oppføringer med kryssreferanser mot UniProtKB/Swiss-Prot-symboler som ikke finnes i testsettet for UniProtKB/Swiss-Prot og 48 oppføringer uten kryssreferanse til UniProtKB/Swiss-Prot. I hele HGNC-databasen er det kun ca 800 ikke unike oppføringer. Dette fører til at testsettet fra HGNC vil inneholde langt flere unike oppføringer enn UniProtKB/Swiss-Prot-testsettet.

For interaksjonsdatabasene er det viktig at testsettene inneholder interaksjoner mellom protein/gen som finnes i testsettene fra UniProtKB/Swiss-Prot og HGNC. De må også inneholde oppføringer der kun en av protein-/genoppføringene finnes i UniProtKB/Swiss-Prot- og HGNC-testsettene, og der ingen av oppføringene finnes. Både IntAct og BioGrid er i sin helhet under 4MB før forbehandling, så disse brukes i sin helhet. Men for at en testing av relasjonsdatabasen skal kunne utføres blir interaksjonsdatabasene splittet i et testsett for første gangs innfylling og et for oppdatering, med overlappende oppføringer.

Etter forbehandling ble testsettet på 14.238 oppføringer for BioGrid og 50.238 oppføringer for IntAct.

Grunnen til at jeg velger å legge begge interaksjonsdatabasene i sin helhet i databasen, er som nevnt at de ikke tar stor plass og det er heller ikke spesielt tidkrevende å legge dem inn.

6.1.2 Testsett for oppdatering av databasen

For å teste oppdatering av relasjonsdatabasen ble et mindre sett, enn for første gangs fylling av databasen, hentet ut fra annoterings- og interaksjonsdatabasene.

Testsettet fra UniProtKB/Swiss-Prot inneholder 50 protein-/genoppføringer. Av disse er 15 oppføringer som ligger i relasjonsdatabasen fra før og har ingen endringer. Fem av oppføringene har endring i protein-/gennavnet, 11 er oppføringer med samme protein-/gennavn som finnes i relasjonsdatabasen, men ikke samme symbol. Dessuten er 19 oppføringer, der verken protein-/gennavnet eller symbolet finnes i relasjonsdatabasen fra før, tatt med. Oppføringene er også plukket ut med kriteriet at de måtte ha en eller flere synonymer, slik at også oppdatering av

tabellen *Synonymer* kan testes. Oppføringene i testsettet inneholder til sammen 30 synonymer, der alle er unike. En del av synonymene finnes er nye og noen finnes også i testsettet for første gangs innfylling. Det er også foretatt noen manuelle endringer i synonymene for å teste endring i synonym navn og endring i tilknytning til protein-/genoppføring.

Testsettet fra HGNC inneholder i likehet med UniProtKB/Swiss-Prot-settet 50 oppføringer, der 15 er oppføringer som allerede finnes i relasjonsdatabasen og har ingen endringer. Fem av oppføringene har endring i protein-/gennavnet, 10 er oppføringer med samme protein-/gennavn som finnes i relasjonsdatabasen, men ikke samme id, og 20 oppføringer der verken protein-/gennavnet eller id finnes i relasjonsdatabasen. Testsettet inneholder også 142 synonymer der noen er nye og noen også finnes i testsettet for første gangs innfylling. Det er i likhet med testsettet for oppdatering av UniProtKB/Swiss-Prot foretatt noen manuelle endringer i synonymene.

Settene for oppdatering av *Interaksjon*-tabellen er den andre delen av de splittede interaksjonsdatabasene, som omtalt i kapittel 6.1.1. Etter forbehandling, i henhold til kapittel 5.2, er testsettene på 17.796 oppføringer for BioGrid, der 500 oppføringer overlapper med settet for første gangs innfylling. For IntAct inneholder testsettet 53.920 oppføringer med 1000 overlappinger.

6.2 Fylling og oppdatering av relasjonsdatabasen

For å teste om relasjonsdatabasen og prototypen som skriver til og leser fra databasen fungerte som den skulle, ble først testsettene fra samtlige databaser lagt til i relasjonsdatabasen. Når all data fra testsettene lå i relasjonsdatabasen, ble testsett for oppdatering brukt til å oppdatere databasen.

For å få testet alle aspekter av relasjonsdatabasen må situasjoner som kan oppstå i databasen identifiseres

6.2.1 Oppføringer i *ProtNavn*-tabellen

For oppføringer i *ProtNavn*-tabellen kan følgende situasjoner oppstå.

Protein-/gennavn finnes i relasjonsdatabasen fra før.

- → Symbol/id finnes fra før: ingen endring i protein-/genoppføringen.
- → Symbol/id finnes ikke fra før: dette er en ny oppføring.

Protein-/gennavn finnes ikke i relasjonsdatabasen fra før.

- → Symbol id/finnes fra før: det har forekommet en endring i protein-/gennavnet.
- → Symbol/id finnes ikke fra før: dette er en ny oppføring.

6.2.2 Oppføringer i *Synonymer*-tabellen

For oppføringer i *Synonymer*-tabellen kan følgende situasjoner oppstå.

Synonym navnet finnes i relasjonsdatabasen fra før.

- → Symbol/id finnes fra før: synonym oppføringen har ikke blitt endret
- → Symbol/id finnes ikke fra før: dette er en ny oppføring

Synonym navnet finnes ikke i relasjonsdatabasen fra før kan:

- → Symbol/id finnes fra før: det kan ha forekommet en endring i synonym navnet, synonymet er ikke lengre tilknyttet protein-/gennavnet eller det kan være et nytt synonym som er blitt tilknyttet protein-/genoppføringen.
- → Symbol/id finnes ikke fra før: dette er en ny oppføring.

6.2.3 Oppføringe i *Kryssref*-tabellen

For oppføringer i *Kryssref*-tabellen kan følgende situasjoner oppstå.

Kryssreferansen finnes i relasjonsdatabasen fra før.

Kryssreferansen finnes ikke i relasjonsdatabasen fra før.

6.2.4 Oppføringer i *Interaksjon*-tabellen

For oppføringer i *Interaksjon*-tabellen kan følgende situasjoner oppstå.

Interaksjonen finnes i relasjonsdatabasen fra før
Interaksjonen finnes ikke i relasjonsdatabasen fra før.

6.3 Gjenfinning av protein-/gennavn og -synonymer i indeksen

For å teste om indeksen fra relasjonsdatabasen, ble nøyaktige søk og delevise søk foretatt på et utvalg termer fra relasjonsdatabasen. De utvalgte søke-termene måtte til samme oppfylle følgende kriterier. Minst en av søketermene må:

- være et tilnærmet søk og det må finnes flere treff i relasjonsdatabasen som vil gi treff på den tilnærmede spørringen.
- være både et protein-/gennavn og et synonym
- ha kryssreferanser og interaksjoner med oppføringer i relasjonsdatabasen.
- ha flere enn et synonym knyttet til seg.

6.4 Test resultat

6.4.1 Fylling av relasjonsdatabasen

proteinNavn	swiss_id	hgnc_id
14-3-3 protein theta	p27348 p68254	NULL
alpha-1-antiproteinase precursor	p34955	NULL
acetoacetyl-coa synthetase	NULL	21298
alpha-1,4-n-acetylglucosaminyltransferase	q9una3	117968
annexin a5	p08758 p17153 p48036 p14668	543 89038496 93302020 32737727
annexin a6	p79134 p51901 p08133	544
argininosuccinate synthetase pseudogene 2	NULL	765

Tabell 4 Eksempler på oppføringer fra *ProtNavn*-tabellen

Tabell 4 viser et eksempel på oppføringer i relasjonsdatabasen etter at både UniProtKB/Swiss-Prot og HGNC er lagt til. Som man kan se av tabellen blir symbol/id til oppføringer med samme protein-/gennavn, alle lagt til i feltet *swiss_id* eller *hgnc_id*. Hvis en oppføring kun finnes i UniProtKB/Swiss-Prot eller HGNC, bli feltet for databasen protein-/gennavnet ikke finnes i, satt til NULL.

synonym	swiss_id	hgnc_id
flj25179	NULL	23336
bai1-associated protein 2		q60437 q9uqb8 q6gmn2
p38	o95433 q04929	NULL
annexin v	p17153 p08758 p48036 p14668	NULL
alpha4gnt	q9una3	17968

Tabell 5 Eksempler på oppføringer fra *Synonymer*-tabellen

Eksempler på oppføringer fra *Synonymer*-tabellen vises i Tabell 5. Dette fungerer på samme måte som ved innfylling av *ProtNavn*-tabellen.

6.4.1.1 Kryssreferanser

swiss_id	hgnc_id
o00116	327
o00253	330
o14678	68
o15218	13708
o15254	121
o15439	55
P08133	544

Tabell 6 Eksempel på oppføringer fra *Kryssref*-tabellen

Tabell 6 viser et utsnitt av tabellen *Kryssref* i relasjonsdatabasen. Denne tabellen består kun av symbol/id. I Tabell 7 og Tabell 8 kan man se eksempler på hvordan man kan bruke Tabell 6 til å hente ut protein-/genoppføringer fra *ProtNavn*, som er kryssreferert. Både Tabell 7 og Tabell 8 viser en enkelt kryssreferanse. Forskjellen er at i Tabell 7 er ikke protein-/gennavnet det samme i begge annoterings-databasene, mens i Tabell 8 er protein-/gennavnet tilsvarende for både UniProtKB/Swiss-Prot og HGNC.

proteinNavn	swiss_id	hgnc_id
alkylglycerone phosphate synthase	NULL	327
alkyldihydroxyacetonephosphate synthase, peroxisomal precursor	o00116	NULL

Tabell 7 Eksempel på kryssreferanse

proteinNavn	swiss_id	hgnc_id
alpha-1,4-n-acetylglucosaminyltransferase	q9una3	17968

Tabell 8 Eksempel på kryssreferanse

6.4.1.2 Synonymer

proteinNavn	swiss_id	hgnc_id	synonym	swiss_id	hgnc_id
acyl-coa synthetase short-chain family member 2	NULL	15814	acyl-coa synthetase short-chain family member 2	q9nr19	NULL
apoptosis-associated tyrosine kinase	NULL	21	apoptosis-associated tyrosine kinase	q6zmq8	NULL
adp/atp translocase 1	p02722 p12235		adp/atp translocase 1	p31167 o22342 p04709	

Tabell 9 Protein-/gennavn som også er synonymer

Tabell 9 er et resultat av en spørring der *synonym* i *Synonymer*-tabellen skal være lik *proteinNavn* i *ProtNavn*-tabellen. Dette viser da protein-/gennavn som finnes i begge tabellene og dermed synonymer som også er selvstendige protein-/gennavn og protein-/gennavn som også er synonymer for andre protein-/gennavn.

6.4.1.3 Interaksjoner

protein1	protein2	db
186	263	HGNC
13633	24041	HGNC
p24666	q9uhj3	SwissProt
p63261	q99962	SwissProt

Tabell 10 Eksempel fra *Interaksjon*-tabellen

Tabell 10 viser eksempler på oppføringer fra tabellen *Interaksjon*. Man kan også se hvilken database interaksjonene refererer til.

protein1	protein2	db	proteinNavn	swiss_id	hgnc_id
p24666	q9uhj3	SwissProt	low molecular weight phosphotyrosine protein phosphatase	p24666	122
p63261	q99962	SwissProt	actin, cytoplasmic 2	p63261	144
o95477	q13884	SwissProt	atp-binding cassette, sub-family a (abc1), member 1	o95477	29
p42684	p01112	SwissProt	tyrosine-protein kinase abl2	p42684	77

Tabell 11 Eksempler på UniProtKB/Swiss-Prot-interaksjoner

proteinNavn	swiss_id	hgnc_id
scm-like with four mbt domains protein 1	q9uhj3	null
sh3-containing grb2-like protein 2	q99962	null
beta-1-syntrophin	q13884	null
gtpase hras precursor	p01112	null

Tabell 12 Eksempler på UniProtKB/Swiss-Prot-interaksjoner

Tabell 11 og Tabell 12 viser hvordan man ved hjelp av interaksjoner som de i Tabell 10 kan finne interaksjoner mellom proteiner. De grå kolonnene viser koblingen mellom oppføringene.

protein1	protein2	db	proteinNavn	swiss_id	hgnc_id
186	263	HGNC	adenosine deaminase	p00813	186
13633	24041	HGNC	adiponectin, c1q and collagen domain containing	q15848	13633

Tabell 13 Eksempler på HGNC-interaksjoner

proteinNavn	swiss_id	hgnc_id
adenylate cyclase 5	o95622	236
adiponectin receptor 2	q86v24	24041

Tabell 14 Eksempler på HGNC-interaksjoner

Tabell 13 og Tabell 14 viser også hvordan man ved hjelp av interaksjoner som de i Tabell 10 kan finne interaksjoner mellom proteiner i HGNC. De grå kolonnene viser koblingen mellom oppføringene.

6.4.2 Oppdatering av relasjonsdatabasen

Ved oppdatering av databasen kan nye oppføringer bli lagt til, et nytt symbol/id kan bli lagt til et protein-/gennavn eller -synonym. Det kan også forekomme at et protein-/gennavn eller synonym navn kan ha blitt endret, noe som fører til sletting

av et symbol/id fra en oppføring eller sletting av en oppføring og en ny oppføring eller symbol/id blir lagt til.

6.4.2.1 Oppdatering av ProtNavn-tabellen

Nytt symbol

proteinNavn	swiss_id	hgnc_id
14-3-3 protein gamma	q5f3w6 p61982	NULL
adrenomedullin receptor	o15218	13708
shikimate kinase	q3svm9 q2g733 q8enm6	NULL
1-acylglycerol-3-phosphate o-acyltransferase 4 (lysophosphatidic acid acyltransferase, delta)	NULL	20885
agouti related protein homolog (mouse)	NULL	330
alanine-glyoxylate aminotransferase	NULL	341
alpha-2-hs-glycoprotein	NULL	349

Tabell 15 Eksempel på oppføringer før oppdatering av relasjonsdatabasen.

proteinNavn	swiss_id	hgnc_id
14-3-3 protein gamma	q5f3w6 p61982 q5rc20 p61983	NULL
adrenomedullin receptor	o15218 p43142 p31392	13708
shikimate kinase	q3svm9 q2g733 q8enm6 o26896 q2ri74 p0a4z3 q9ccs5 q741j8 p0a4z2 q5fad3 o50467	NULL
agouti related protein homolog (mouse)	NULL	330 33000
alanine-glyoxylate aminotransferase	NULL	334411341
alpha-2-hs-glycoprotein	NULL	349 943

Tabell 16 Samme oppføringer som i Tabell 15 etter oppdatering av relasjonsdatabasen.

Tabell 15 og Tabell 16 viser hvordan symbol/id-er kan bli lagt til og fjernet fra oppføringer i relasjonsdatabasen ved en oppdatering.

Endring i protein-/gennavn

Tabell 17 og Tabell 18 viser til sammen et eksempel på de forskjellige situasjonene som kan oppstå ved oppdatering av *ProtNavn*-tabellen når det er endring i protein-/gennavnene med symbol/id som allerede finnes i databasen. Endring i protein-/gennavn kan føre til følgende endringer i *ProtNavn*-tabellen:

Symbol/id blir fjernet fra den opprinnelige oppføringen. Hvis da ikke oppføringen lengre har noen symbol/id fra verken UniProtKB/Swiss-Prot eller HGNC knyttet til seg, blir oppføringen slettet. Inneholder fortsatt oppføringen andre symbol/id, blir oppføringen fortsatt liggende i relasjonsdatabasen. Hvis protein-/gennavnet ikke finnes i relasjonsdatabasen fra før, blir det lagt til en ny oppføring. Hvis protein-/gennavnet finnes i relasjonsdatabasen fra før, blir kun symbol/id lagt til i oppføringen.

proteinNavn	swiss_id	hgnc_id
14-3-3 protein 3	p93209	NULL
antirepressor protein 1	p19655	NULL
brain-specific angiogenesis inhibitor 1-associated protein 2	q60437 q9uqb8 q6gmn2	NULL
14-3-3 protein 6	p93211	NULL
atp-binding cassette, sub-family a (abc1), member 2	NULL	32
atp-binding cassette, sub-family b (mdr/tap), member 4	NULL	45
alpha-1,4-n-acetylglucosaminyltransferase	q9una3	17968

Tabell 17 Eksempel på oppføringer før oppdatering av relasjonsdatabasen

proteinNavn	swiss_id	hgnc_id
14-3-3 protein 33	p93209	NULL
antirepressor core protein 1	p19655	NULL
brain-specific angiogenesis inhibitor 1-associated protein 2	q9uqb8 q6gmn2	NULL
brain-specific angiogenesis inhibitor 1-associated beta protein 2	q60437	NULL
atp-binding cassette, sub-family ade (abc1), member 2	NULL	32
annexin a5	p17153 p08758 p48036 p14668 p93211	89038496 32737727 93302020 543
arylacetamide deacetylase-like 1	q6piu2	29260 45
alpha-1,4-n-acetylglucosaminyltransferase	q9una3	NULL
alpha-1,4-n-acetylglucosaminyltransferasefrase	NULL	17968

Tabell 18 Hvordan oppføringene i Tabell 17 kan se ut etter oppdatering.

Tabell 17 og Tabell 18 viser at oppføringen med symbol `*p93209*` har endret navn fra `*14-3-3 protein 3*` til `*14-3-3 protein 33*`. Dette fører til at oppføringen `*14-3-3 protein 3*` i Tabell 17 ikke lengre har noen symbol/id knyttet til seg og blir dermed slettet fra relasjonsdatabasen, mens oppføringen `*14-3-3 protein 33*` finnes ikke i relasjonsdatabasen fra før og blir lagt til. Det samme gjelder for oppføringene `*antirepressor protein 1*` og `*atp-binding cassette, sub-family a (abc1) , member 2*`. Oppføringen `*brain-specific angiogenesis inhibitor 1-associated protein 2*` og `*alpha-1,4-n-acetylglucosaminyltransferase*`.

I Tabell 17 er som man kan se av Tabell 18 ikke fjernet fra relasjonsdatabasen, selv om protein-/gennavnet er endret og symbol/id dermed er fjernet. Grunnen til dette er at oppføringen fortsatt har symbol fra UniProtKB/Swiss-Prot og/eller id fra HGNC knyttet til seg. Dermed kan ikke oppføringen fjernes fra relasjonsdatabasen. Protein-/gennavnene i oppføringene med UniProtKB/Swiss-Prot-symbol `*p93211*` og med HGNC-id `*45*` i Tabell 17 har blitt endret til protein-/gennavn som allerede ligger i relasjonsdatabasen, protein-/genoppføringene `*annexin a5* og *arylacetamide deacetylase-like 1*`. Dermed blir symbol/id lagt til den aktuelle oppføringen, som man kan se i Tabell 18 og oppføringene fra Tabell 17 blir slettet fra relasjons-databasen, fordi de ikke lengre har noe symbol/id knyttet til seg.

6.4.2.2 Oppdatering av Synonymer-tabellen

Nytt symbol/id

synonym	swiss_id	hgnc_id
sk	p63600 q3svm9 q2g733 q8enm6	NULL
am-r	o15218	NULL

Tabell 19 Eksempler oppføringer i *Synonymer*-tabellen før oppdatering

synonym	swiss_id	hgnc_id
sk	p63600 q3svm9 q2g733 q8enm6 o26896 q2ri74 p0a4z3 q9ccs5 q741j8 p0a4z2 q5fad3 o50467	NULL
am-r	o15218 p43142 p31392	NULL

Tabell 20 Eksempel på hvordan oppføringer tilsvarende Tabell 19 i *Synonymer*-tabellen kan se ut etter oppdatering

Tabell 19 og Tabell 20 viser hvordan symboler fra UniProtKB/Swiss-Prot har blitt lagt til oppføringen ved oppdatering av relasjonsdatabasen.

Endring i synonym navn

Som tidligere nevnt er endring i synonym navn vanskelig å oppdage. Endring i synonymoppføringer kan derfor være en faktisk endring i synonym navn eller en ny oppføring. Noe som kan føre til at synonym oppføringer i relasjonsdatabasen ikke lengre er tilknyttet protein-/genoppføringen.

Testingen av oppdatering gikk ut fra dette ut på å sjekke om nye synonymer ble lagt til og hvorvidt symbol/id ble lagt til der synonym navnet allerede eksisterer i relasjonsdatabasen. Dessuten ble det testet om det for synonymer som ikke lengre var tilknyttet et gitt protein-/genoppføring, ble slettet symbol/id eller hele oppføringen.

synonym	swiss_id	hgnc_id
bai1-associated protein 2	q60437 q9uqb8 q6gmn2	NULL
bai-associated protein 2	q60437	17968
insulin receptor substrate p53	q60437	NULL
irsp53	q60437	NULL
alanine trna ligase 1	NULL	20
cytoplasmic	NULL	20

Tabell 21 Synonymer for oppføringene med symbol/id q60437 og 25230

synonym	swiss_id	hgnc_id
bai1-associated protein 2	q60437 q9uqb8 q6gmn2	NULL
bai-associated protein 2	NULL	17968
insulin receptor substrate protein of 53 kda	q60437	NULL
p58/p53	q60437	NULL
alanine trna ligase 1234	NULL	17968, 20
cytoplasmic	NULL	20

Tabell 22 Oppdaterte synonymer

Tabell 21 viser synonymene for oppføringen med UniProtKB/Swiss-Prot-symbol `*q60437*` og HGNC-id `*20*` før oppdatering av relasjonsdatabasen. Tabell 22 viser synonymene for de samme protein-/genoppføringen etter oppdatering av relasjons-databasen. Som vi kan se er det ingen endring i oppføringene `*bai1-associated protein 2* og *cytoplasmic*`. Synonymet `*bai-associated protein 2*` er ikke lengre tilknyttet protein-/genoppføringen med symbol `*q60437*`. Symbolet blir dermed slettet fra oppføringen i Tabell 22, men siden oppføringen er tilknyttet andre protein-/genoppføringer, blir den ikke slettet.

Når det gjelder synonymene `*insulin receptor substrate p53*`, `*irsp53*` og `*alanine trna ligase 1*` blir alle disse slettet fra relasjonsdatabasen fordi de etter oppdatering ikke lengre er tilknyttet noen protein-/genoppføring. Dette kommer av at de ikke lengre er synonymer for oppføringene `*q60437*` og `*20*` i annoteringsdatabasene.

Som man kan se av Tabell 22 er `*insulin receptor substrate protein of 53 kda og p58/p53*` og `*alanine trna ligase 1234*` nye synonymer, som ikke er tilknyttet protein-/genoppføringene fra før. Disse blir da enten lagt til som nye oppføringer i *Synonymer*-tabellen eller symbol/id blir lagt til hvis synonymet allerede finnes. Det sistnevnte gjelder for synonymet `*alanine trna ligase 1234*`, som allerede ligger i relasjonsdatabasen og er tilknyttet protein-/genoppføringen med HGNC-id `*17968*`.

6.4.2.3 Oppdatering av Kryssref og Interaksjons-tabellen

Det er en av to muligheter ved oppdatering av *Kryssref* og *Interaksjons*-tabellen. Kryssreferansen/interaksjonen kan enten ligge i relasjonsdatabasen fra før, eller den blir lagt til som en ny oppføring. Det finnes ikke muligheter for å sjekke om kryssreferanser/interaksjoner er fjernet fra annoterings- eller interaksjons-

databasene, siden dette vil bli alt for ressurskrevende. Det vil da, som tidligere nevnt, være mer gunstig å tømme *Kryssref*/*Interaksjon*-tabellen og fylle de på nytt.

6.4.2.4 Resultat av søk i indeksen

Protein-/gennavn*et annexin-b10-13* har følgende informasjon knyttet til seg:

Symbol/id: p22465

Synonym:

annexin-10

Interaksjoner:

null

Kryssreferanser:

null

Protein-/gennavn*et annexin-b11* har følgende informasjon knyttet til seg:

Symbol/id: q9vxg4

Synonym:

null

Interaksjoner:

q9vxg4 p19109-2 UniProt

q9vxg4 q24141 UniProt

q9vxg4 q9v6i8 UniProt

q9vxg4 q9v755 UniProt

q8msx2 q9vxg4 UniProt

Kryssreferanser:

null

Synonymet *annexin v* er tilknyttet protein-/gennavn med følgende symbol/id:

p17153

p08758

p48036

p14668

Figur 6 Figuren fortsetter på neste side

Synonymet *annexin vi* er tilknyttet protein-/gennavn med følgende symbol/id:

p79134

p51901

p08133

Synonymet *annexin vii* er tilknyttet protein-/gennavn med følgende symbol/id:

p20072

Synonymet *annexin xi* er tilknyttet protein-/gennavn med følgende symbol/id:

p27214

p33477

Synonymet *annexin-10* er tilknyttet protein-/gennavn med følgende symbol/id:

p22465

Figur 7 Eksempel på jokertegn søk i indeksen der søketermen er *annexin

Som man kan se av Figur 7 vil et søk på termen `*annexin*`, der et jokertegn brukes for å gi tilnærmet treff, gi et resultat på 11 protein-/gennavn og -synonymer. I resultatet vises først protein-/genoppføringene med tilhørende symbol/id, synonymer, interaksjoner og kryssreferanser. Oppføringen med UniProtKB/Swiss-Prot-symbolet `*q9vxg4*` har flere interaksjoner med andre protein/gen. Resultatet viser også synonymer som starter med termen `*annexin*` og symbol/id, fra UniProtKB/Swiss-Prot og HGNC, for protein-/gennavnene de er tilknyttet.

Protein-/gennavnet *acyl-coa thioesterase 2* har følgende informasjon knyttet til seg:

Symbol/id: 18431

Synonym:

mte1

zap128

pte2

mitochondrial acyl-coa thioesterase 1

Interaksjoner:

null

Kryssreferanser:

p49753 18431

Synonymet *acyl-coa thioesterase 2* er tilknyttet følgende protein-/gennavn:

p49753

Figur 8 Eksempel på søkeresultat ved søk i indeksen på termen *acyl-coa thioesterase 2*

Figur 8 viser resultat for spørringen med den nøyaktige termen navnet `*acyl-coa thioesterase*`. Denne termen er oppført som både protein-/gennavn og -synonym i relasjonsdatabasen. Resultatet viser derfor både synonymer, interaksjoner og kryssreferanser for protein-/gennavnet og symbol/id for protein-/gennavn som synonymet er tilknyttet. Som vi kan se finnes det flere synonymer tilknyttet oppføringen og det finnes en kryssreferanse til oppføringen med id `*18431*` i HGNC.

6.4.3 Oppsummering test

Ut fra testresultatene kan man se at relasjonsdatabasen løser kombinasjons-mulighetene beskrevet i kapittel 4.2.1. Flere oppføringer kan ha same synonym, flere oppføringer kan ha samme protein-/gennavn og proteinnavn kan være synonym for en annen oppføring. Indekseringen av databasen løser problemet med

tilnærmet like oppføringer ved å bruke tilnærmet søk. Dessuten viser testen av indeksen tilleggsinformasjon fra relasjonsdatabasen enkelt kan hentes ut.

Testen sier ikke noe om hvor lang tid det tar å fylle relasjonsdatabasen eller hvordan den vil fungere når man legger til hele utgaver av annoterings- og interaksjonsdatabasen. Grunnen til dette er at tida det tar å fylle relasjonsdatabasen varierer veldig. Dette har med at relasjonsdatabasen ikke ligger lokalt og dermed kommer det an på Internetkoblingen hvor lang tid prototypen bruker for å fylle databasen. Når det gjelder test med hele utgaver av annoterings- og interaksjons-databaser var ikke dette hensiktsmessig på grunn av størrelsen på databasene, spesielt UniProtKB/Swiss-Prot.

7 Diskusjon

Motivasjonen for denne oppgaven har vært å se på mulighetene for å samle flere annoterings- og interaksjonsdatabaser og dermed forbedre mulighetene for bedre identifisering av protein-/genforekomster i abstrakter. For å kunne undersøke dette er forskjellige vurderinger foretatt og en prototype har blitt utviklet for å teste og underbygge vurderingene.

I dette kapittelet blir vurderingene foretatt og prototypen implementert diskutert. Det blir også for hver del av prototypen foreslått muligheter for videre arbeid.

7.1 Annoterings- og interaksjonsdatabasene

UniProtKB/Swiss-Prot-databasen er enkel å laste ned og hente ut data fra. Med HGNC finnes det flere problemer, som i dette tilfellet ble løst manuelt. For det første oppstår det problemer fordi HGNC ikke tilbyr filnedlasting av databasen på en slik måte som gjør at den er enkel å hente ut. HGNC kan som tidligere omtalt kun hentes ut i tabulator-delt tekst eller html. I denne oppgaven ble databasen lastet ned kun med de feltene som var nyttige. Ved en automatisk nedlasting, hvis det er mulig, vil man sannsynligvis måtte laste ned hele innholdet til HGNC, noe som vil føre til at prototypen vil bruke lengre tid på å hente ut informasjonen som skal brukes.

Det oppstod også problemer med å hente ut informasjon fra HGNC-fila fordi en del oppføringer inneholder tomme felter, som blir ”usynlige” ved uthenting av data. Dermed ble det vanskelig å lage en metode for å hente ut data fra fila. Dette ble løst ved manuelt å sette inn ’null’ i de tomme feltene, men ved en videreutvikling av prototypen, mot en mer automatisk applikasjon, vil ikke dette være gunstig. Formatmessig er derfor ikke HGNC mest gunstig å bruke, men ble likevel valgt på grunn av innholdet og dermed muligheten for å teste relasjonsdatabasen.

Når det gjelder interaksjonsdatabasene er tabulator-delt tekstfil godt egnet, fordi de inneholder interaksjoner der ingen av de to aktuelle feltene kan være tomme. Dessuten er det færre felt i disse databasene og dermed enklere å hente ut informasjon og unngå store mengder unødvendig informasjon.

Det viste seg at det optimale filformatet for å samle flere annoteringsdatabaser er XML. Grunnen til dette er at en XML-fil enkelt kan analyseres og man henter ut kun den informasjonen som er forespurt. Dessverre er det slik at mange interaksjons- og annoteringsdatabaser ikke finnes i XML-format. Det går likevel an å bruke disse, selv om metodene for uthenting av data kan bli noe mer komplisert. Men finnes annoterings- og interaksjonsdatabasen i XML-format, anbefales dette.

Slik prototypen er utviklet her blir tilpassede filer fra de forskjellige annoterings- og interaksjonsdatabasene lastet ned manuelt. Grunnen til dette var for det første at det ville være lite gunstig å bruke hele utgaven av databasen til testing. Det ville også ta veldig lang tid å fyll relasjonsdatabasen med hele utgaven av de forskjellige annoterings- og interaksjonsdatabasene. For videreutvikling av prototypen vil det være naturlig at data fra de forskjellige databasene hentes ut automatisk. Da vil man kunne sette applikasjonen til å oppdatere relasjonsdatabasen og indeksen med gitte tidsintervaller. Dette krever for øvrig mye minne og at applikasjonen er web-basert.

7.2 Forbehandling

I teoridelen av oppgaven ble mange forskjellige forbehandlingssteg beskrevet. Det er imidlertid svært få av dem var aktuelle for dette formålet. Mye av grunnen til dette er at de i utgangspunktet er utviklet for forbehandling av dokumenter som skal indekseres i forbindelse med vanlig informasjonsgjenfinning. Dessuten inneholder svært mange protein-/gennavn og -synonymer tegn, som i vanlig informasjons-gjenfinning er vanlig å fjerne, men som ikke bør fjernes i denne sammenheng. Siden denne oppgaven kun begrenser seg til utviklingen av en samling av annoterings- og interaksjonsdatabaser er det vanskelig å vurdere hvordan oppføringene i relasjonsdatabasen skal forbehandles. Det ble derfor valgt å beholde oppføringene mest mulig som de var i utgangspunktet. Det eneste som ble gjort var å gjøre alle tegn om til små bokstaver, samt fjerne lange oppføringer, slik at identifisering/søk er være uavhengig av det. Som nevnt forutsetter dette fraværet av forbehandling på oppføringene i relasjonsdatabasen, at tilsvarende gjøres på abstraktene som skal brukes til identifisering. En annen grunn til at det ikke ble foretatt noen spesiell forbehandling av oppføringene, er hensynet til resultatet

brukeren vil få ut. Ved en eventuell forbehandling med fjerning av forskjellige tegn og stoppord, kan man risikert at resultatet brukeren får ut er et protein-/gennavn eller -synonym som er endret og kan være ugjenkjennelig. Dette ville være til svært liten nytte for brukeren.

Vurderingen av hvilke forbehandlings steg som skal utføres, kommer i stor grad an på formålet med identifiseringen. For en videreutvikling av prototypen må det vurderes hvordan dataene som legges i relasjonsdatabasen skal forbehandles ut fra hvordan man ønsker å forbehandle abstraktene og hva resultatet av identifiseringen skal brukes til.

7.3 Relasjonsdatabasen

Relasjonsdatabasen ble utformet på grunnlag av kriteriene i kapittel 4.2. Den største utfordringen i forbindelse med samling av flere annoteringsdatabaser er at samme protein/gen kan ha forskjellige navn eller navnevarianter i forskjellige annoteringsdatabaser. Dette ble løst ved hjelp av kryssreferanser mellom databasene. Problemet med dette er at det er vanskelig å vite om databasene er oppdatert og i tilfelle hvilken av dem som er det. Det kan med andre ord forekomme at en kryssreferanse er feil på grunn av at den ene annoteringsdatabasen er oppdatert, men ikke den andre. Noe annet som også kan forekomme på grunn av oppdatering, eller mangel på det, er at kryssreferanser som egentlig burde vært i annoteringsdatabasene ikke er der. For eksempel består HGNC av en rekke oppføringer uten kryssreferanse til UniProtKB/Swiss-Prot. Hvis da UniProtKB/-Swiss-Prot har blitt oppdatert med nye eller endrede oppføringer, kan det finnes oppføringer i HGNC som burde kryssreferere til UniProtKB/Swiss-Prot men ikke gjør det. Det et lite å gjøre med dette problemet, men for å få en best mulig felles samling, er det viktig å bruke annoteringsdatabaser som oppdateres jevnlig. Som nevnt i kapittel 2.7.1 er det blitt tilført eller oppdatert 312.089 oppføringer, i perioden 01.04.07-01.05.07. Dette viser at UniProtKB/Swiss-Prot oppdateres jevnlig.

Problemer ved at databasene ikke er oppdatert i forhold til hverandre, kan også oppstå mellom interaksjons– og annoteringsdatabasene. Hvis en oppføring i UniProtKB/Swiss-Prot har blitt slettet, slått sammen med en annen oppføring eller på annen måte blitt endret, kan interaksjonen som ligger i interaksjonsdatabasen

være utriktig. Dermed kan brukeren få feilinformasjon om interaksjoner, men det kan også skje at man ikke får informasjon om interaksjoner som finnes fordi interaksjonsdatabasen ikke er oppdatert. Det er derfor i likhet med annoteringsdatabasene viktig å bruke interaksjonsdatabaser som er oppdatert.

Lagring av symbol/id i forbindelse med protein-/gennavn og -synonymer ble forsøkt gjort på enklest mulig måte i relasjonsdatabasen. Tabellen *ProtNavn* og *Synonymer* inneholder derfor i tillegg til protein-/gennavnet eller -synonymet, et felt for hver annoteringsdatabase. Alternativet til dette ville vært dobbeltlagring av protein-/gennavn og -synonymer eller et felt for hvert symbol/id. Det førstnevnte ville føre til en unødvendig stor relasjonsdatabase. Det sistnevnte ville være tilnærmet umulig fordi man ikke ville vist hvor mange felt man trengte og dessuten fått svære mange tomme felt i databasen.

Ulempen med en løsning der flere symbol/id ligger i samme felt, kun skilt ved mellomrom, er at det ikke er fullt så enkelt å bruke feltet. I prototypen finnes det flere metoder for å skille symbol/id-er i samme felt fra hverandre. Dette gjør naturlig nok prototypen mer kompleks. Det kan også oppstå problemer hvis andre annoteringsdatabaser blir lagt til og disse har mellomrom i sine symbol/id. Dette kan løses ved å bruke et egnet tegn som skille istedenfor mellomrom. Det er med andre ord viktig å være klar over at det kan oppstå problemer i forbindelse med symbol/id i andre annoteringsdatabaser, selv om denne løsningen fungerer godt i denne sammenhengen.

7.3.1 Oppdatering

En oppdatering av relasjonsdatabasen foregår i stor grad på samme måte som ved første gangs innfylling. Testresultatene viser at relasjonsdatabasen oppdaterer seg hensiktsmessig, men det er likevel et par haker ved det.

Ved oppdatering av *Synonymer*-tabellen er det *tilnærmet* umulig å sjekke hvorvidt en oppføring er ny eller endret. Det er for øvrig mulig for en samling synonymer å sjekke hvorvidt et synonym er fjernet fra en oppføring annoteringsdatabasen og at man dermed også må fjerne symbol/id fra synonymet i relasjonsdatabasen. Dette utføres i prototypen og sørger for at synonymer som ikke lengre er tilknyttet en protein-/genoppføring blir fjernet, samt at protein-/genoppføring ikke har synonymer knyttet til seg som er fjernet fra oppføringen.

Oppdatering både i *Kryssref*- og *Interaksjon*-tabellen foregår ved at man sjekker om kryssreferansen eller interaksjonen ligger der fra før og hvis den ikke gjør det legges den til. Det finnes ingen metode i prototypen for å finne kryssreferanser eller interaksjoner som ikke finnes lengre. Ulempen med dette er at man ved søk mot indeksen, kan få ut et resultat som viser kryssreferanser og interaksjoner som ikke egentlig eksisterer lengre. Den enkleste løsningen på dette er å tømme *Kryssref*- og *Interaksjon*-tabellene med jevne mellomrom, for så å fylle dem på nytt.

Det har vært viktig i denne oppgaven å utvikle en mest mulig generell tilnærmelse til en samling av flere annoterings- og interaksjonsdatabaser. Relasjonsdatabasen er derfor konstruert på en slik måte at flere annoterings- og interaksjonsdatabaser enn de som er brukt her, enkelt kan legges til. Dette er gjort ved at kun de mest vanlige og nødvendige feltene fra annoterings- og interaksjonsdatabasene er tatt med. Samtidig er det lagt vekt på de forskjellige situasjonene som kan oppstå ved samling av flere databaser.

En mulig utvidelse av relasjonsdatabasen er å legge til et felt med forkortelser, slik at også disse kan identifiseres i abstrakter og bidra til bedre informasjons-gjenfinning. Forkortelser i biomedisinsk tekst er omtalt i kapittel 2.1.1.4, men har ikke vært en del av hovedfokuset i denne oppgaven. Det er også en ide å utvide tabellene *ProtNavn* og *Synonymer* til å inneholde et dato-felt der dagens dato legges inn hvis oppføringen blir oppdatert. Dette vil kunne gjøre oppdateringen av indeksen mer effektiv, fordi man da kan plukke ut oppføringer oppdatert etter en viss dato og oppdatere indeksen med disse og ikke hele relasjonsdatabasen. Dette vil være ressursbesparende, hvis relasjonsdatabasen inneholder store mengder data.

7.4 Indeksering

Indeksering av relasjonsdatabasen gjøres i dette tilfellet hver gang prototypen starter opp og indeksen ligger kun i minnet. Dette fungerer bra og er raskere enn å lagre indeksen på disk hver gang prototypen starter, som hadde vært alternativet i dette tilfellet. Grunnen til dette er for det første naturlig nok at testsettene for relasjonsdatabasene ikke er så store. Dessuten er det raskere å lage indeks i minnet enn på disk. Ingen av disse to alternativene er likevel optimale for en

relasjonsdatabase der de fullstendige utgavene av annoteringsdatabasene er lagt til. Grunnen er at prototypen vil bli treg fordi en indeks må bygges hver gang prototypen starter opp. Dessuten vil indeksen bli for stor til å kunne holdes i minnet og må dermed lagres midlertidig på disk og være tregere. Et annet argument for å lagre en indeks mer permanent på disk, er at indeksen kan oppdateres istedenfor å bygges på nytt, samt at en oppdatering av relasjonsdatabasen kan foregå uten at det påvirker indeksen. Det optimale er derfor at en permanent indeks lagres sammen med relasjonsdatabasen. Dette forutsetter at applikasjonen er web-basert, slik at en bruker enkelt får tilgang til indeksen og tilleggsdata fra relasjons-databasen.

Indeksen inneholder kun to felter. Et felt for selve protein-/gennavnet eller -synonym og et felt for å merke om det er et protein-/gennavn eller -synonym. Dette fungerte bra fordi resultatet av søk mot indeksen viser hvorvidt treffet er fra *ProtNavn* eller *Synonymer*-tabellen. Dessuten gir det større muligheter for å hente ut kun den informasjonen som er nyttig for brukeren fra relasjonsdatabasen.

En videreutvikling av indekseringen vil som nevnt være å lagre den mer permanent på disk. Dessuten vil det være naturlig å implementere en metode for oppdatering av indeksen.

7.5 Søk og resultat

For å kunne sjekke om relasjonsdatabasen og indeksen fungerte til sitt formål ble søk mot indeksen implementert i prototypen. Både nøyaktige og tilnærmede søk kan gjøres mot indeksen. Dette viser at relasjonsdatabasen og indeksen fungerer til sitt formål, nemlig identifisering av protein-/gennavn og -synonym forekomster i abstrakter. Et resultat gir også ut tillegges informasjon fra databasen, som gjør at interaksjoner kan identifiseres.

Siden det egentlige formålet med indeksen er identifisering av protein-/gen-forekomster i abstrakter vil en naturlig videreutvikling være å teste indeksen mot en applikasjon som henter ut potensielle kandidater for protein-/gennavn fra abstrakter. Dette vil gi en videre pekepinne på hvordan relasjonsdatabasen og indekseringen

av den kan videreutvikles, for å fungere optimalt ved identifisering av protein-/gen-forekomster.

7.6 Tidligere arbeid

Det finnes svært få tilgjengelige artikler om tidligere tilnærmelser til samling av annoterings- og interaksjonsdatabaser for bedre identifisering av protein-/gen-forekomster og -interaksjoner mellom disse i biomedisinsk litteratur. Men blant annet Moon og Singh [19] har vært inne på tanken om en domenespesifikk ordbok. Mye av grunnen til at lite forskning er gjort på dette området kan være de store forskjellene i annoterings- og interaksjonsdatabasene. Svært mange av dem har forskjellig innhold og forskjellig struktur. Dessuten er det stor forskjell på hva av innholdet som er tilgjengelig og hvordan det er tilgjengelig. Disse faktorene til sammen gjør at det å skulle samle mange annoterings- og interaksjonsdatabaser vil være svært ressurskrevende og dermed kanskje ikke spesielt lønnsomt med tanke på hva man får igjen for det.

Med denne oppgaven er man et steg nærmere en slik ordbok, men det gjenstår mange utfordringer. Å lage en samling i form av en relasjonsdatabase er, som vist i denne oppgaven, mulig og kan også være et bidrag til å forbedre identifisering av protein-/genforekomster ved hjelp av ordbok. De aller største utfordringene ligger i de annoterings- og interaksjonsdatabasene som skal brukes i en slik samling. De må for det første være fritt tilgjengelige og det må være mulig innholds og formatmessig å kunne samle dem. Et problem som påpekes av Narayanaswamy m.fl.[23] er at det ikke finnes nøyaktige overensstemmelser om hva som utgjør et protein-/gennavn. Denne oppgaven har vist hvordan man kan komme et steg nærmere en løsning på dette problemet, ved å kombinere flere annoteringsdatabaser. Til tross for dette vil det sannsynligvis være gunstig å kombinere en slik samlet ordbok med andre tilnærmelser, både regel- og maskinbaserte, for å optimalisere identifiseringen av protein-/genforekomster i abstrakter.

Kim m.fl.[22] bruker ordbok i sin regel-baserte tilnærmelse som identifiserer interaksjoner på bakgrunn av kontraster. Krysslinking av kontrastproteinnavnene fra litteraturen med oppføringer i en standard proteindatabase, som for eksempel

UniProtKB/Swiss-Prot, viser at mulighetene for en kombinering av flere tilnærmelser er til stede og kan forbedre identifisering av protein-/genforekomster og -interaksjoner.

8 Oppsummering

I denne oppgaven har jeg sett på mulighetene for en felles ordbok for identifisering av protein-/gennavn og interaksjoner i artikkelsammendrag. Hensikten med en felles ordbok er å samle flere annoterings- og interaksjonsdatabaser, for dermed å kunne forbedre identifiseringen. Vurderingene foretatt i oppgaven er støttet av en prototype, som viser hvordan dette kan gjøres i praksis.

Annoteringsdatabasene brukt er UniProtKB/Swiss-Prot og HGNC(HUGO Gene Nomenclature Committee). Disse ble valgt ut som testdatabaser på grunn av at de begge inneholdt protein-/gennavn, id/symbol og synonymer, samt at de hadde kryssreferanser til hverandre. Dessuten var de lett tilgjengelige. Interaksjons-databasene som ble brukt, IntAct og BioGrid, består av interaksjoner basert på symbol/id fra henholdsvis UniProtKB/Swiss-Prot og HGNC. Det passet derfor svært godt å bruke disse i en felles samling.

De utvalgte annoterings- og interaksjonsdatabasen fungerte bra til dette formålet og viser at en felles samling er mulig. Samtidig vil det sannsynligvis kunne oppstå problemer på flere områder når det gjelder å legge til andre annoterings- og interaksjonsdatabaser enn de som er brukt her. Blant annet har veldig mange av databasene som finnes forskjellig innhold og struktur. Det er også forskjell på hvor enkelt det er å hente ut informasjon fra dem både når det gjelder hvor åpne de er og på hvilken måte man kan hente ut informasjon fra dem. Dette er sannsynligvis noe av hovedgrunnen til at det ikke er foretatt noe særlig forskning innenfor dette området.

I prototypen utviklet i forbindelse med denne oppgaven ble det valgt å bruke relasjonsdatabase for lagring av en felles ordbok. Dette var mest hensiktsmessig av alternativene som ble vurdert fordi det i denne sammenheng var viktig å lagre relasjonene innad i annoterings- og interaksjonsdatabasene og relasjoner mellom databasene. Dessuten unngår man i stor grad dobbeltlagring av data, samt at indeksering og gjenfinning av data i relasjonsdatabasen er forholdsvis enkelt. Testresultatene viste at strukturen til relasjonsdatabasen fungerte bra med de

utvalgte annoterings- og interaksjonsdatabasene. Både første gangs innfylling og oppdatering ble testet på bakgrunn av identifiserte situasjoner som kunne oppstå.

Relasjonsdatabasen ble forsøkt utviklet med tanke på at flere annoterings- og interaksjonsdatabaser kunne legges til. Dette burde være mulig med denne løsningen, men kommer også an på hvilke andre databaser det er aktuelt å legge til.

For at identifisering av protein-/genforekomster og -interaksjoner i artikkel-sammendrag skal bli rask og mest mulig effektiv, blir relasjonsdatabasen indeksert. Indeksen bygges hver gang prototypen starter og ligger i minnet. Det er hensiktsmessig i denne sammenheng på grunn av mengden data, men som en del av videre arbeid vil det være hensiktsmessig å lagre indeksen på disk og gjøre applikasjonen web-basert.

Indeksen inneholder protein-/gennavn og -synonymer. Dette gjør at identifiseringen kan foregå raskere fordi man ikke trenger å søke gjennom all data i relasjonsdatabasen. Samtidig kan oppføringen i indeksen brukes til å hente ut tilleggsinformasjon fra relasjonsdatabasen, som symbol/id, interaksjoner og kryssreferanser mellom annoteringsdatabasene. Dermed kan brukeren få nyttig tilleggsinformasjon i tillegg til protein-/gennavn og -synonymer som identifiseres i artikkelsammendrag. I prototypen er det mulig å foreta tekstsøk mot indeksen og dermed få ut all informasjon om protein-/gennavn og -synonymer. Testingen viser at dette fungerer bra, også med tilnærmet søk. Videre arbeid i den sammenheng er å teste indeksen mot et system som henter ut protein-/gennavn- og -synonymkandidater fra artikkelsammendrag, for å se hvordan indeksen fungerer til identifisering.

8.1 Konklusjon

Oppgaven viser at en felles samling av annoterings- og interaksjonsdatabaser for identifisering av protein-/genforekomster og -interaksjoner i artikkelsammendrag er fullt mulig og kan være hensiktsmessig. Grunnen til dette er at man får en mer fullstendig samling av protein-/gennavn, samtidig som man også har informasjon om interaksjoner i samme samling og dermed kan foreta flere identifiserings-

oppgaver samtidig. Dette vil gi brukeren mer utfyllende og samlet. Samtidig er det en stor utfordring at annoterings- og interaksjonsdatabasene er så forskjellige. Det må derfor vurderes om det med tanke på ressursbruk og det man kan få igjen for det, er hensiktsmessig og forsette arbeidet med å utvikle en slik samling.

8.2 Videre arbeid

Det er mange ting å ta tak i når det gjelder videre arbeid. For det første bør indeksen testes mot en applikasjon som henter ut potensielle kandidater for protein-/gennavn fra abstrakter. Dette vil gi en videre pekepinne på hvordan relasjons-databasen og indekseringen av den kan videreutvikles for å fungere optimalt ved identifisering av protein-/genforekomster.

Når det gjelder videreutvikling av relasjonsdatabasen, kan protein-/gennavn-forkortelser legges til, slik at også disse kan identifiseres i abstrakter, og dermed bidra til bedre gjenfinning. Det kan også være en mulighet å utvide tabellene *ProtNavn* og *Synonymer* til å inneholde et dato-felt der dagens dato legges inn hvis oppføringen blir oppdatert. Dette vil kunne gjøre oppdateringen av indeksen mer effektiv.

Hovedmålet med en felles ordbok for identifisering av protein-/genforekomster og -interaksjoner er at den skal være en oppdatert og effektiv samling. Dette vil kreve automatisering av datauthenting fra annoterings- og interaksjonsdatabasene og en automatisk oppdatering av relasjonsdatabasen og indeksen. Man bør derfor se på mulighetene for å gjøre applikasjonene web-basert med en sentralisert relasjons-database og indeks.

A Referanser

1. Lacroix, Z.a.T.C., *Bioinformatics : managing scientific data*. 2003, San Francisco: Morgan Kaufmann Publishers. 441.
2. NCBI. *Just the Facts: A basic introduction to the science underlying NCBI resources*. A Science Primer 2007 [cited 2007 24.04]; Available from: http://www4.ncbi.nlm.nih.gov/about/primer/bioinformatics.html.
3. McNaught, J.a.A., Sophia, *Text mining for biology and biomedicine*. 2006, Boston: Artech House. XI, 286 s.
4. Kunnskapsforlaget. *ordnett.no*. 2007 [cited 2007 30.04]; Available from: http://www.ordnett.no.
5. Baeza-Yates, R.a.R.-N., Berthier, *Modern information retrieval*. ACM Press books. 1999, New York: ACM Press. XX, 513 s.
6. Tan, P.-N.e.a., *Introduction to data mining*. 2006, Boston: Pearson Addison Wesley. XXI, 769 s.
7. Bunescua, R.e.a., *Comparative experiments on learning information extractors for proteins and their interactions.* Artificial Intelligence in Medicine, 2004. **33**: p. 16.
8. Bray, T.e.a., *Extensible Markup Language (XML) 1.0 (Fourth Edition).* W3C Recommendation, 2006.
9. Weiss, S.M., *Text mining : predictive methods for analyzing unstructured information*. 2005, New York: Springer. XII, 237 s.
10. Date, C.J., *An introduction to database systems*. 2004, Boston: Pearson/Addison Wesley. XXVII, 983, 22 s.
11. Chen, H.e.a., ed. *Medical Informatics - Knowledge management and data mining in biomedicine*. Springer's intergrated series in information systems. 2005, Springer Science
12. Liang, Y.D., *Introduction to Java programming: fundamentals first*. 2007, Upper Saddle River, N.J.: Pearson/Prentice Hall. XXV, 678 s.
13. Gospodnetic, O.a.H., Erik, *Lucene in action*. 2005, Greenwich, CT: Manning. xxxiv, 421 s.
14. Medicine, U.S.N.L.o. *Fact sheet - Medline*. 2006 [cited 2007 28.04]; Available from: http://www.nlm.nih.gov/pubs/factsheets/medline.html.
15. UniProt, C. *UniProt - The universal protein resource*. 2007 2006 [cited 2007 23.01]; Available from: http://www.ebi.uniprot.org/index.shtml.
16. HUGO, G.N.C. *HGNC* 2007 16.02.2007 [cited 2007 18.04.2007]; Available from: http://www.gene.ucl.ac.uk/nomenclature/data/gdlw_columndef.html#gd_hgnc_id.
17. Kerrien, S.e.a., *IntAct—open source resource for molecular interaction data.* Nucleic Acids Research, 2006. **35**.
18. Stark, C.e.a., *BioGRID: a general repository for interaction datasets.* Nucleic Acids Research, 2005. **34**.
19. Moon, N.a.S., Rahul *Experiments in Text-Based Mining and Analysis of Biological Information from MEDLINE on Functionally-Related Genes.* IEEE, 2005.
20. Yamamotoy, K.e.a., *Protein Name Tagging for Biomedical Annotation in Text.* 2003.

21. Tsuruokazy, Y.a.T., Jun'ichi, *Boosting Precision and Recall of Dictionary-Based Protein Name.* 2003.
22. Kim, J.-j.e.a., *BioContrasts: extracting and exploiting protein–protein contrastive relations from biomedical literature.* Bioinformatics, 2006. **22**: p. 9.
23. Narayanaswamy, M.e.a., *A biological named entity recognizer.* Pac Symp Biocomput., 2003. **8**.
24. Soni, A., *Protein Interaction Extraction from Medline Abstracts Using Conditional Random Fields.* 2006.
25. Bunescua, R.C.o.R.J.M., *Using Biomedical Literature Mining to Consolidate the Set of Known Human Protein-Protein Interactions* 2005: p. 8.
26. Daraselia, N.e.a., *Extracting human protein interactions from MEDLINE using a full-sentence parser.* Bioinformatics, 2004. **20**: p. 7.

B Annoteringsdatabaser

Eksemplene fra annoteringsdatabasene er manuell forbehandling som beskrevet i oppgaven.

Eksempeloppføring fra UniProtKB/Swiss-Prot, lastet ned i XML-format

Det som er uthevet, er dataene fra en UniProtKB/Swiss-Prot oppføring som blir brukt i denne sammenheng.

```
<entry dataset="Swiss-Prot" created="1995-11-01" modified="2007-03-06" version="42">
 <accession>P42652</accession>
 <accession>Q8L5E2</accession>
 <name>14334_SOLLC</name>
 <protein>
  <name>14-3-3 protein 4</name>
  <name>PBLT4</name>
 </protein>
 <gene>
  <name type="primary">TFT4</name>
 </gene>
 <organism key="1">
  <name type="full">Solanum lycopersicum (Tomato) (Lycopersicon esculentum)</name>
  <dbReference type="NCBI Taxonomy" id="4081" key="2"/>
  <lineage>
   <taxon>Eukaryota</taxon>
   <taxon>Viridiplantae</taxon>
   <taxon>Streptophyta</taxon>
```

```
      <taxon>Embryophyta</taxon>
      <taxon>Tracheophyta</taxon>
      <taxon>Spermatophyta</taxon>
      <taxon>Magnoliophyta</taxon>
      <taxon>eudicotyledons</taxon>
      <taxon>core eudicotyledons</taxon>
      <taxon>asterids</taxon>
      <taxon>lamiids</taxon>
      <taxon>Solanales</taxon>
      <taxon>Solanaceae</taxon>
      <taxon>Solanoideae</taxon>
      <taxon>Solaneae</taxon>
      <taxon>Solanum</taxon>
      <taxon>Lycopersicon</taxon>
    </lineage>
  </organism>
  <reference key="3">
    <citation type="journal article" date="1995" name="Biochim. Biophys. Acta" volume="1263" first="67" last="70">
      <title>Two cDNA clones encoding 14-3-3 homologs from tomato fruit.</title>
      <authorList>
        <person name="Laughner B."/>
        <person name="Lawrence S.D."/>
        <person name="Ferl R.J."/>
      </authorList>
      <dbReference type="PubMed" id="7632735" key="4"/>
```

```
      <dbReference type="MEDLINE" id="95359205" key="5"/>
      <dbReference type="DOI" id="10.1016/0167-4781(95)00092-U" key="6"/>
    </citation>
    <scope>NUCLEOTIDE SEQUENCE [MRNA].</scope>
    <source>
      <strain>cv. Ailsa Craig</strain>
      <tissue>Fruit</tissue>
    </source>
  </reference>
  <reference key="7">
    <citation type="journal article" date="1999" name="Plant Physiol." volume="119" first="1243" last="1250">
      <title>Fusicoccin, 14-3-3 proteins, and defense responses in tomato plants.</title>
      <authorList>
        <person name="Roberts M.R."/>
        <person name="Bowles D.J."/>
      </authorList>
      <dbReference type="PubMed" id="10198082" key="8"/>
      <dbReference type="MEDLINE" id="99214454" key="9"/>
      <dbReference type="DOI" id="10.1104/pp.119.4.1243" key="10"/>
    </citation>
    <scope>NUCLEOTIDE SEQUENCE [MRNA].</scope>
    <source>
      <strain>cv. Moneymaker</strain>
      <tissue>Leaf</tissue>
    </source>
```

```
</reference>
<comment type="subunit" status="Potential">
  <text>Homodimer</text>
</comment>
<comment type="similarity">
  <text>Belongs to the 14-3-3 family</text>
</comment>
<dbReference type="EMBL" id="L29150" key="11">
  <property type="protein sequence ID" value="AAA99431.1"/>
  <property type="molecule type" value="mRNA"/>
</dbReference>
<dbReference type="EMBL" id="AJ504807" key="12">
  <property type="protein sequence ID" value="CAD43308.1"/>
  <property type="molecule type" value="mRNA"/>
</dbReference>
<dbReference type="PIR" id="S57272" key="13">
  <property type="entry name" value="S57272"/>
</dbReference>
<dbReference type="UniGene" id="Les.2473" key="14"/>
<dbReference type="SMR" id="P42652" key="15">
  <property type="residue range" value="4-237"/>
</dbReference>
<dbReference type="HSSP" id="P93343" key="16">
  <property type="pdb accession" value="1O9E"/>
</dbReference>
```

```
<dbReference type="InterPro" id="IPR000308" key="17">
  <property type="entry name" value="14-3-3"/>
</dbReference>
<dbReference type="Gene3D" id="G3DSA:1.20.190.20" key="18">
  <property type="entry name" value="14-3-3"/>
  <property type="match status" value="1"/>
</dbReference>
<dbReference type="PANTHER" id="PTHR18860" key="19">
  <property type="entry name" value="14-3-3"/>
  <property type="match status" value="1"/>
</dbReference>
<dbReference type="Pfam" id="PF00244" key="20">
  <property type="entry name" value="14-3-3"/>
  <property type="match status" value="1"/>
</dbReference>
<dbReference type="PRINTS" id="PR00305" key="21">
  <property type="entry name" value="1433ZETA"/>
</dbReference>
<dbReference type="ProDom" id="PD000600" key="22">
  <property type="entry name" value="14-3-3"/>
  <property type="match status" value="1"/>
</dbReference>
<dbReference type="SMART" id="SM00101" key="23">
  <property type="entry name" value="14_3_3"/>
  <property type="match status" value="1"/>
```

```
</dbReference>
<dbReference type="PROSITE" id="PS00796" key="24">
  <property type="entry name" value="1433_1"/>
  <property type="match status" value="1"/>
</dbReference>
<dbReference type="PROSITE" id="PS00797" key="25">
  <property type="entry name" value="1433_2"/>
  <property type="match status" value="1"/>
</dbReference>
<feature type="chain" description="14-3-3 protein 4" id="PRO_0000058684">
  <location>
    <begin position="1"/>
    <end position="260"/>
  </location>
</feature>
<feature type="sequence conflict" description="(in Ref. 2)" ref="7">
  <original>D</original>
  <variation>E</variation>
  <location>
    <position position="96"/>
  </location>
</feature>
<feature type="sequence conflict" description="(in Ref. 2)" ref="7">
  <original>H</original>
  <variation>Y</variation>
```

<location>

<position position="125"/>

</location>

</feature>

<sequence length="260" mass="29299" checksum="955E5BE81765CBB9" modified="1995-11-01" version="1">

MADSSREENVYLAKLAEQAERYEEMIEFMEKVAKTADVEELTVEERNLLS

VAYKNVIGARRASWRIISSIEQKEESRGNEDHVNTIKEYRSKIEADLSKI

CDGILSLLESNLIPSASTAESKVFHLKMKGDYHRYLAEFKTGTERKEAAE

NTLLAYKSAQDIALAELAPTHPIRLGLALNFSVFYYEILNSPDRACNLAK

QAFDEAISELDTLGEESYKDSTLIMQLLRDNLTLWTSDNADDVGDDIKEA

SKPESGEGQQ

</sequence>

</entry>

Eksempel på oppføringer fra HGNC, lastet ned i tab-delt tekst format

HGNC ID	Approved Name	Aliases	Name Aliases	UniProt ID
HGNC:168	ARP1 actin-related protein 1 homolog B, centractin beta (yeast)	null	null	P42025
HGNC:172	activin A receptor, type IB	ALK4, SKR2, ActRIB	null	P36896
HGNC:173	activin A receptor, type IIA	ACTRII	null	P27037
HGNC:175	activin A receptor type II-like 1	HHT2, ALK1, HHT	null	P37023
HGNC:24104	aspartoacylase (aminocyclase) 3	HCBP1, MGC9740, ACY-3	null	Q96HD9
HGNC:180	acylphosphatase 2, muscle type	null	null	P14621
HGNC:148	actin, gamma pseudogene 3	null	null	null
HGNC:215	ADAM metallopeptidase domain 8	CD156, MS2	null	P78325
HGNC:188	ADAM metallopeptidase domain 10	kuz, MADM, HsT18717, CD156c	null	O14672
HGNC:147	actin, gamma pseudogene 2	null	null	null
HGNC:195	ADAM metallopeptidase domain 17 (tumor necrosis factor, alpha, converting enzyme)	cSVP, CD156B	null	P78536
HGNC:197	ADAM metallopeptidase domain 19 (meltrin beta)	MLTNB	null	Q9H013
HGNC:200	ADAM metallopeptidase domain 21	ADAM31	null	Q9UKJ8
HGNC:19706	ADAMTS-like 4	DKFZP434K1772	null	Q6UY14
HGNC:225	adenosine deaminase, RNA-specific	ADAR1	null	P55265
HGNC:227	adenosine deaminase, RNA-specific, B2 (RED2 homolog rat)	RED2, hRED2, ADAR3	null	Q9NS39
HGNC:29957	arginine decarboxylase	ODC-p, ODC1L, KIAA1945	ornithine decarboxylase like	Q96A70
HGNC:19039	aarF domain containing kinase 2	MGC20727	null	Q7Z695

HGNC:21738	aarF domain containing kinase 5	FLJ35454	null	Q3MIX3
HGNC:238	adenylate cyclase 7	KIAA0037	null	P51828
HGNC:240	adenylate cyclase 9	null	null	O60503
HGNC:242	adenylate cyclase activating polypeptide 1 (pituitary) receptor type I	PACAPR	null	P41586
HGNC:244	adducin 2 (beta)	ADDB	null	P35612
HGNC:253	alcohol dehydrogenase 5 (class III), chi polypeptide	ADH-3, ADHX	null	P11766
HGNC:256	alcohol dehydrogenase 7 (class IV), mu or sigma polypeptide	ADH-4	null	P40394
HGNC:30576	acireductone dioxygenase 1	SIPL, MTCBP-1, ARD, APL1, FLJ10913, HMFT1638	membrane-type 1 matrix metalloproteinase cytoplasmic tail binding protein-1	Q9BV57
HGNC:13633	adiponectin, C1Q and collagen domain containing	ACRP30, AdipoQ, apM1, GBP28, adiponectin	adipose most abundant gene transcript 1	Q15848
HGNC:24041	adiponectin receptor 2	PAQR2, ACDCR2	null	Q86V24

C Interaksjonsdatabaser

Eksemplene fra interaksjonsdatabasene. Eksemplene er ikke forbehandlet, som beskrevet i oppgaven.

Eksempel på oppføringer fra IntAct, lastet ned i tab-delt tekst format

ID interactor A	ID interactor B
uniprotkb:P19235	uniprotkb:O00145
uniprotkb:Q13838	uniprotkb:O00148
uniprotkb:Q92793	uniprotkb:O00150
uniprotkb:O43707	uniprotkb:O00151
genbank_protein_gi:4505953	uniprotkb:P14866
uniprotkb:O00141	genbank_protein_gi:6175096
uniprotkb:Q13418	genbank_protein_gi:13514813
uniprotkb:Q8TA85	uniprotkb:O00151
uniprotkb:P54253	uniprotkb:O00154
uniprotkb:P62993	uniprotkb:O00159
uniprotkb:Q6P5Y1	uniprotkb:O00159
uniprotkb:Q99759	uniprotkb:O00159
uniprotkb:Q9H9Y6	uniprotkb:O00159
uniprotkb:P31946	genbank_protein_gi:37551908
uniprotkb:Q01844	uniprotkb:O00160
uniprotkb:Q6PSM0	uniprotkb:O00165
uniprotkb:Q99759	uniprotkb:O00165
uniprotkb:Q49AR1	uniprotkb:O00170
uniprotkb:Q07817	uniprotkb:O00198
uniprotkb:Q07817-1	uniprotkb:O00198
uniprotkb:Q92843	uniprotkb:O00198
uniprotkb:P49888	uniprotkb:O00204
uniprotkb:P05067	uniprotkb:O00213
uniprotkb:P54253	uniprotkb:O00213
uniprotkb:Q06481	uniprotkb:O00213
uniprotkb:P50591	uniprotkb:O00220

uniprotkb:O00221	uniprotkb:O00221
uniprotkb:Q01201	uniprotkb:O00221
uniprotkb:Q04206-2	uniprotkb:O00221
uniprotkb:Q04864	uniprotkb:O00221
uniprotkb:Q00653-2	uniprotkb:O00231
uniprotkb:Q9Y4K3	uniprotkb:O00231
uniprotkb:Q9Y4K3	uniprotkb:O00232
genbank_protein_gi:40038273	uniprotkb:P57088
uniprotkb:P31946	genbank_protein_gi:45503708
uniprotkb:Q03164	uniprotkb:O00255
uniprotkb:Q9Y5Z7	uniprotkb:O00255
uniprotkb:Q99496	uniprotkb:O00257
uniprotkb:Q16514	uniprotkb:O00268
uniprotkb:Q9NRL3	uniprotkb:O00273
uniprotkb:O60925	uniprotkb:O00291
uniprotkb:P42858	uniprotkb:O00291
uniprotkb:P46783	uniprotkb:O00291
uniprotkb:Q13352	uniprotkb:O00291
uniprotkb:P50552	uniprotkb:O00308
uniprotkb:Q01844	uniprotkb:O00308
uniprotkb:Q9NR12	uniprotkb:O00308
uniprotkb:P25205	uniprotkb:O00311
uniprotkb:P38936	uniprotkb:O00311

Eksempel på oppføringer fra BioGrid, lastet ned i tab-delt tekst format

INTERACTOR_A	INTERACTOR_B
HGNC:100	HGNC:100
HGNC:99	HGNC:100
HGNC:99	HGNC:100
HGNC:2231	HGNC:10000
HGNC:2231	HGNC:10000
HGNC:2231	HGNC:10000
HGNC:9113	DAQB-126H3.2
HGNC:9113	DAQB-126H3.2
HGNC:9113	DAQB-126H3.2
HGNC:9113	DAQB-126H3.2
HGNC:9113	DAQB-126H3.2
HGNC:9113	DAQB-126H3.2
HGNC:9113	DAQB-126H3.2
HGNC:7391	HGNC:10003
HGNC:7391	HGNC:10003
HGNC:9008	HGNC:10003
HGNC:9008	HGNC:10003
HGNC:9008	HGNC:10003
HGNC:30304	HGNC:10004
HGNC:30304	HGNC:10004
HGNC:30519	HGNC:10004
HGNC:4401	HGNC:10004
HGNC:4689	HGNC:10004
HGNC:11027	HGNC:10006
HGNC:1682	HGNC:10006
HGNC:4703	HGNC:10006
HGNC:492	HGNC:10006
HGNC:492	HGNC:10006
HGNC:5347	HGNC:10006
HGNC:5347	HGNC:10006
HGNC:14068	HGNC:1001

HGNC:14068	HGNC:1001
HGNC:4544	HGNC:10012
HGNC:4544	HGNC:10012
HGNC:4545	HGNC:10012
HGNC:10012	HGNC:10013
HGNC:10012	HGNC:10013
HGNC:10012	HGNC:10013
HGNC:10012	HGNC:10013
HGNC:10012	HGNC:10013
HGNC:10012	HGNC:10013
HGNC:9937	HGNC:10013
HGNC:10021	HGNC:10019
HGNC:11280	HGNC:10019
HGNC:11280	HGNC:10019
HGNC:11904	HGNC:10019
HGNC:11905	HGNC:10019
HGNC:11916	HGNC:10019
HGNC:11916	HGNC:10019
AC002420.1	HGNC:3676
AC002420.1	HGNC:3676
AC7.2	WormBase:WBGene00002335
AC7.2	WormBase:WBGene00012339
B0024.14	B0024.14
B0024.14	C37C3.6
B0024.14	F48E8.1
B0024.14	WormBase:WBGene00000063
B0024.14	WormBase:WBGene00000083
B0024.14	WormBase:WBGene00001244
B0024.14	WormBase:WBGene00004476
B0024.14	WormBase:WBGene00009081
B0024.14	WormBase:WBGene00009259