Bio Data Management

Kapitel 9

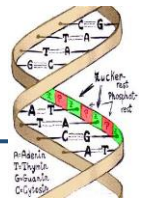
Ontologie-Matching

Wintersemester 2014/15
Dr. Anika Groß
Universität Leipzig, Institut für Informatik, Abteilung Datenbanken
http://dbs.uni-leipzig.de

UNIVERSITÄT LEIPZIG

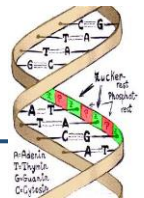
Vorläufiges Inhaltsverzeichnis

1. Motivation und Grundlagen
2. Bio-Datenbanken
3. Datenmodelle und Anfragesprachen
4. Modellierung von Bio-Datenbanken
5. Sequenzierung und Alignments
6. Genexpressionsanalyse
7. Annotationen
8. Datenintegration: Ansätze und Systeme
9. Ontologie-Matching
10. Versionierung von Datenbeständen



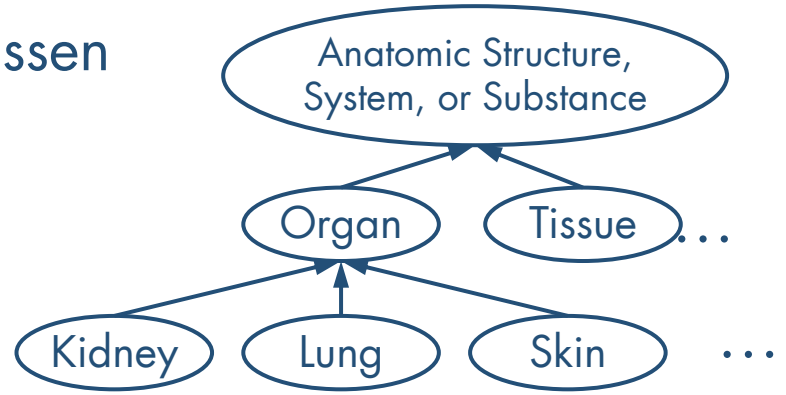
Inhalt

- **Ontologie-Matching**
  - Motivation, Definition, Probleme
  - Klassifikation von Match-Ansätzen
- **Large-Scale Ontology Matching**
  - Probleme bei großen Ontologien
  - Ansätze zum zeiteffizienten Matching von Ontologien



Ontologien

- Strukturierte Repräsentation von Wissen
  Konzepte, Beziehungen
- Sehr große Ontologien



Anatomie

MGI

FMA
University of Washington
School of Medicine

Uber Anatomy Ontology

Medizin

SNOMED CT

NCIthesaurus

Chemie

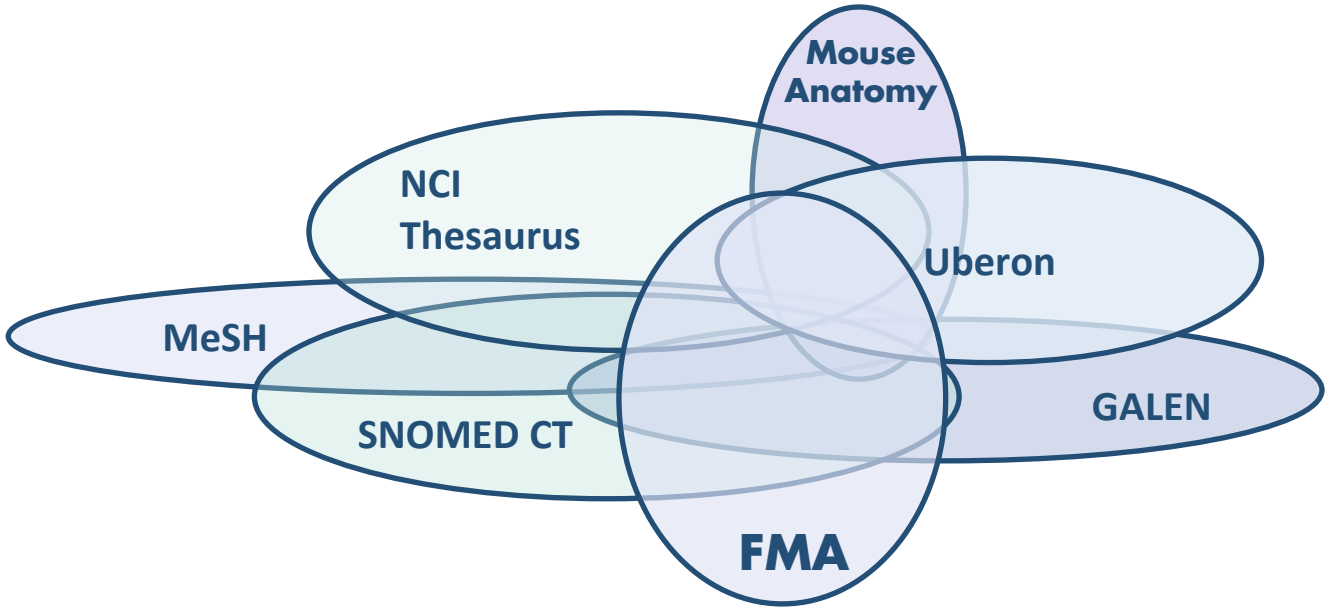
ChEBI

Molekular-biologie

GO
the Gene Ontology

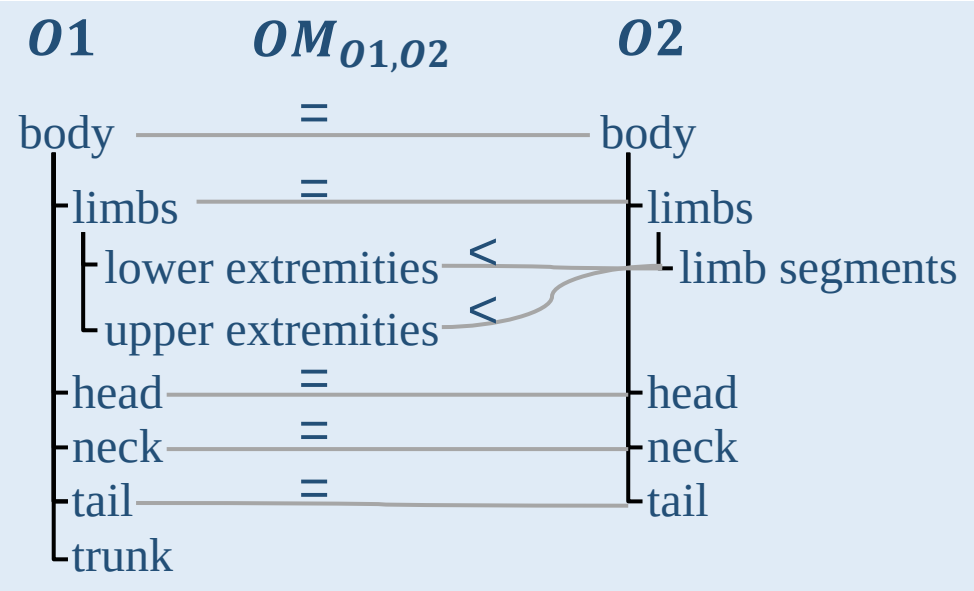
Ontologiemappings

- Überlappende Ontologien → Erstellung von Mappings



Ontologiemappings

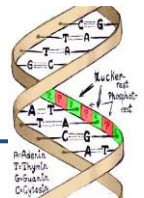
- Überlappende Ontologien → Erstellung von Mappings
- Ontologiemapping: Menge semantischer Korrespondenzen zwischen den Konzepten verschiedener Ontologien



- Manuelle Identifikation oder (semi-) automatische Match-Verfahren

Anwendungen

- Navigation / Browsing / quellübergreifende Analysen
  - Navigation zwischen Datenquellen (z.B. Linked Data)
  - Übertragung von Analyseergebnissen (z.B. Maus → Mensch)
- Datentransformation
  - Integration von Daten
- Anfrageverarbeitung (Umformulierung von Anfragen)
  - P2P Netzwerke, Web Service Komposition, …
- Datenmigration bei Ontologieänderungen
  - Ontology Evolution
- Ontologie-Merging (Uberon, UMLS)



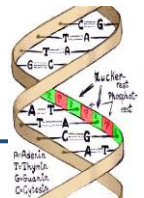
Ontology Matching – Definition

Definition

Ontology Matching is the process of finding relationships or correspondences between entities (concepts, categories) of two different ontologies *(Euzenat, Shvaiko: Ontology Matching, 2007)*

- Identifikation semantischer Korrespondenzen zwischen verschiedenen Ontologien

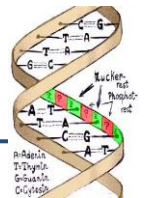
→ **Prozess!** (d.h. Algorithmus, Methode, Strategie) nicht das Ergebnis selbst



Ontology Matching – Eingabe/Ausgabe

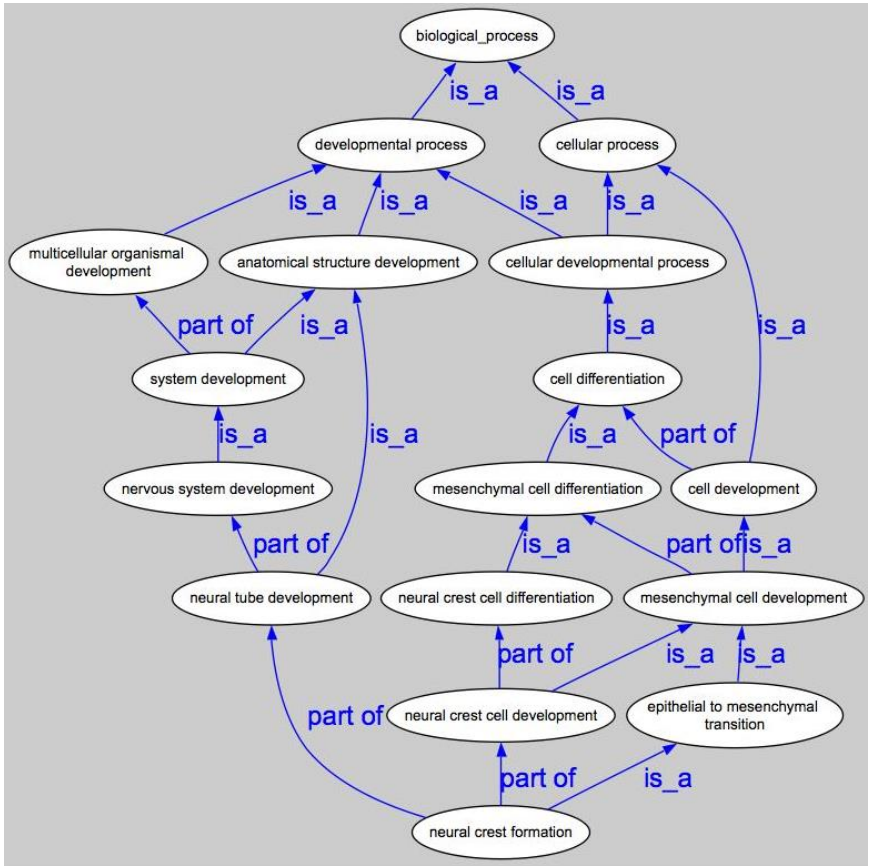
- *Eingabe*
  - Zwei Ontologien $O1$ und $O2$
  - Evtl. Instanzen, Annotationen zu $O1/O2$
  - Evtl. weiteres Hintergrundwissen
    - Domänenwissen wie Wörterbücher, Abkürzungsverzeichnisse, …
    - Andere Terminologien in der gleichen Domäne
- *Ausgabe*
  - **Mapping** (Alignment) zwischen $O1$ und $O2$, d.h. eine Menge von Korrespondenzen zwischen semantisch ähnlichen Konzepten / Kategorien

$$M_{O1,\,O2} = \{(c1, c2, sim) \mid c1 \in O1, c2 \in O2, sim \in [0,1]\}$$



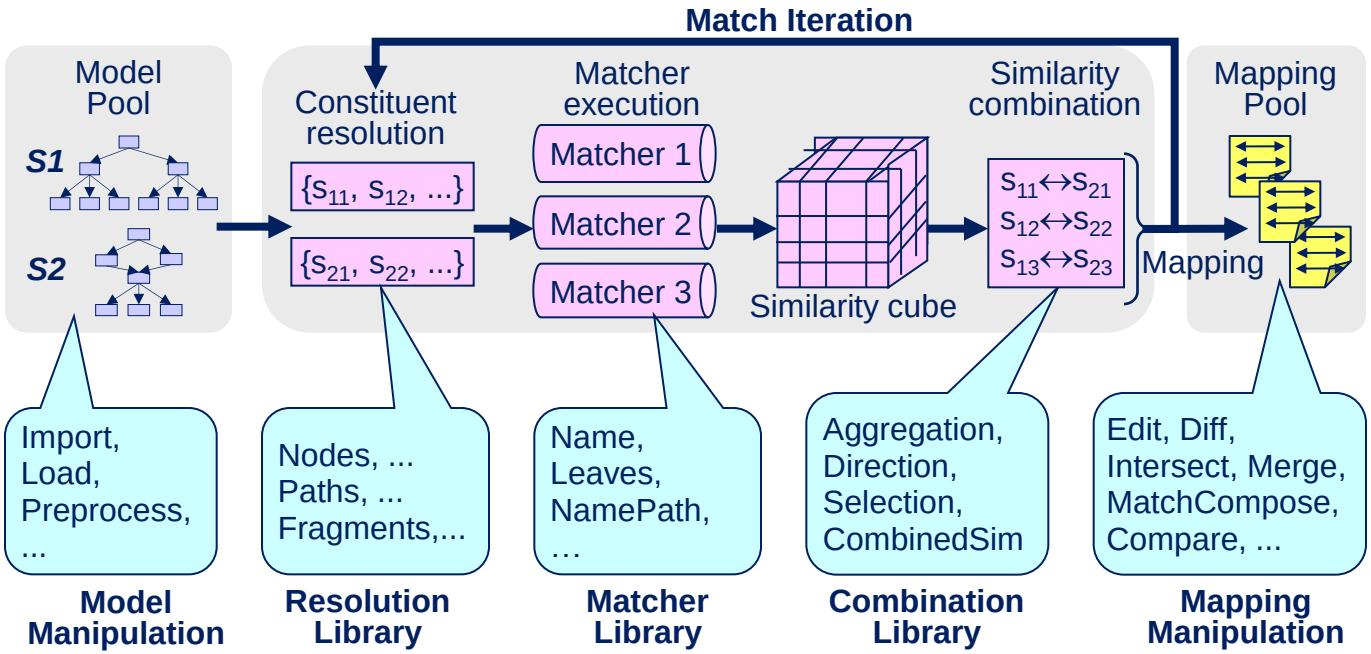
Ontology Matching – Probleme

- **Groß und unübersichtlich**
  - > 10.000 Konzepte
  - Tiefe Hierarchien
- **Unabhängige Entwickelung**
  - Vers. Arten der Heterogenität
    - Fremdsprachenproblem
    - Unbekannte Synonyme, Homonyme
    - Verwendete Abkürzungen
- **Schwierigkeit**
  - Alle Korrespondenzen finden
  - Vermeidung falscher Korrespondenzen

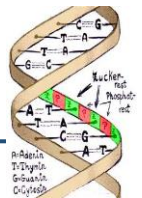


Matching-Prozess

- Iterativer Prozess bestehend aus verschiedenen Aktionen, Match, Kombination
- Bsp. COMA

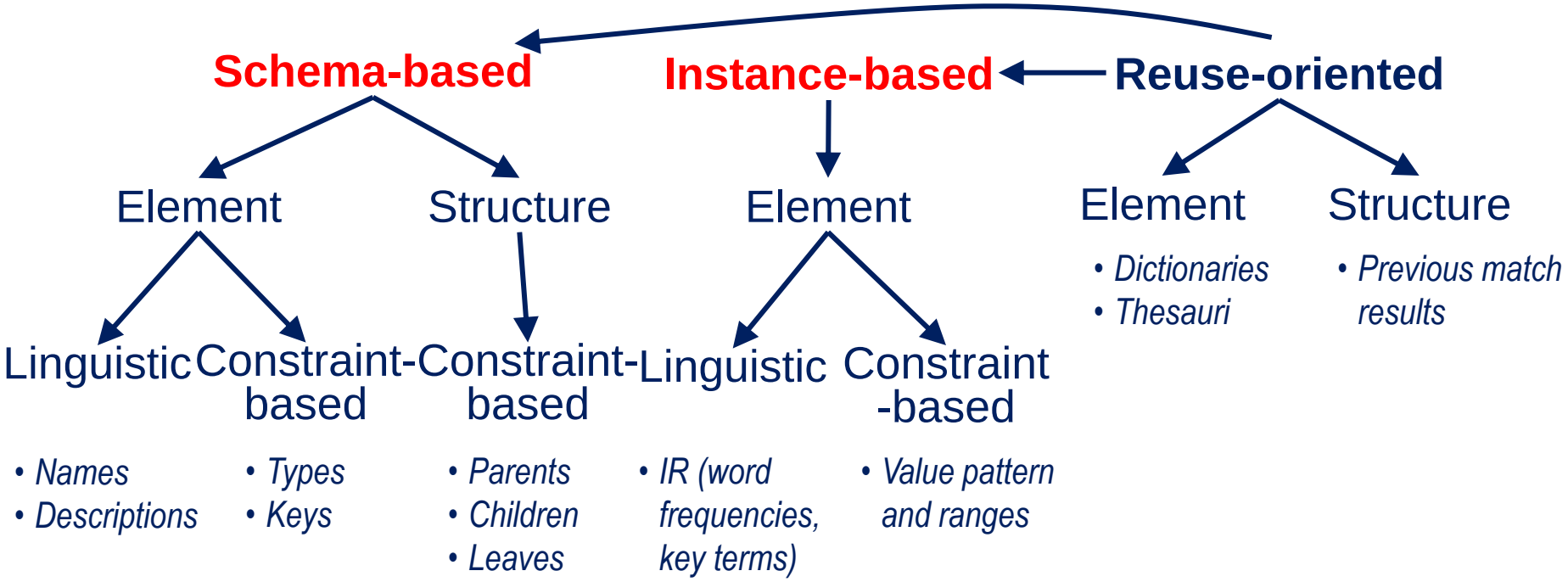


Quelle: Aumüller, Do, Maßmann, Rahm: Schema and Ontology matching with COMA++. Proc. 24th SIGMOD Conf. 2005

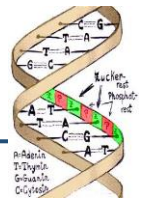


Automatische Ansätze zum Schema Matching

- Klassifikation nach Rahm, Bernstein: *A Survey of Approaches to Automatic Schema Matching. VLDB Journal, 2001*

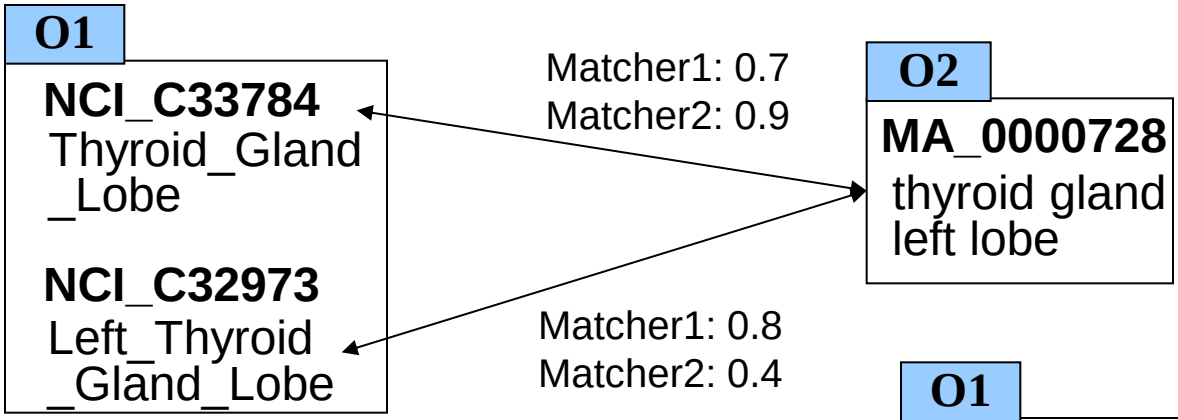


- Kombinierte Ansätze: *Composite* vs. *Hybrid*
  - *Hybrid*: ein Algorithmus wendet mehrere Verfahren an
  - *Composite*: Kombination der Ergebnisse mehrerer Verfahren

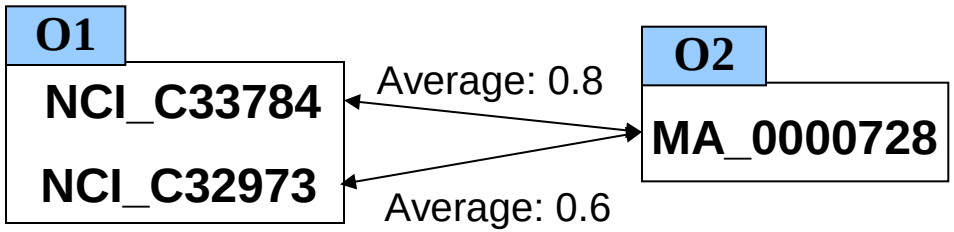


Kombination von Match-Ergebnissen

1. Matcher-Ausführung



2. Aggregation



3. Selektion

Max1

O2 concepts	*O1 concepts*	*Sim*
MA_0000728	NCI_C33784	0.8

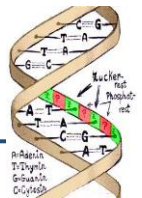
Threshold(0.5)

O2 concepts	*O1 concepts*	*Sim*
MA_0000728	NCI_C33784	0.8
MA_0000728	NCI_C32973	0.6

Large-Scale Ontology Matching

Problem

- Trotz Automatisierung teilweise enormer Rechenaufwand für Erzeugung von Mappings erforderlich
- **Grundkomplexität** ist typischerweise quadratisch **O(m·n)** (mit m = #Konzepte O1, n = #Konzepte in O2)
  - Vergleiche alle Konzepte aus O1 mit allen aus O2 (kartesisches Produkt)
  - Vervielfachung durch Anwendung mehrerer Matcher in Workflows sowie durch Komplexität des Matchers selbst
  - *Beispiel:* NCI Thesaurus (90.000) – Foundational Model of Anatomy (70.000) → 90.000 · 70.000 = 6,3 Mrd. Vgl.
- OAEI (**O**ntology **A**lignment **E**valuation **I**nitiative)
  - Teilweise mehrere Stunden bzw. Tage
- Für interaktive Systeme unzureichend

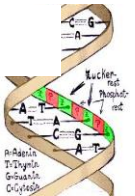


Large-Scale Ontology Matching

Generelle Lösungsansätze

- Suchraumreduzierung vor Matcher-Anwendung
  - Unnötige Vergleiche eliminieren („pruning“)
  - Nur ähnliche Teilbereiche (Partitionen) vergleichen
- Paralleles Matching
  - Durchführung der Matcher-Anwendung auf mehreren CPUs / Rechenknoten bzw. Cloud-Infrastrukturen
- Self-Tuning von Match-workflows
  - Erlernen von optimalen Konfigurationen anstatt manuelles Setting
- Wiederverwendung / Reuse
  - Bereits bekannte Mappings zur Neuberechnung heranziehen

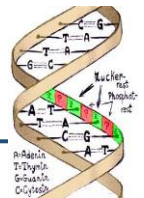
Rahm: Towards Large-Scale Schema and Ontology Matching, Springer, 2011.



Suchraumreduktion

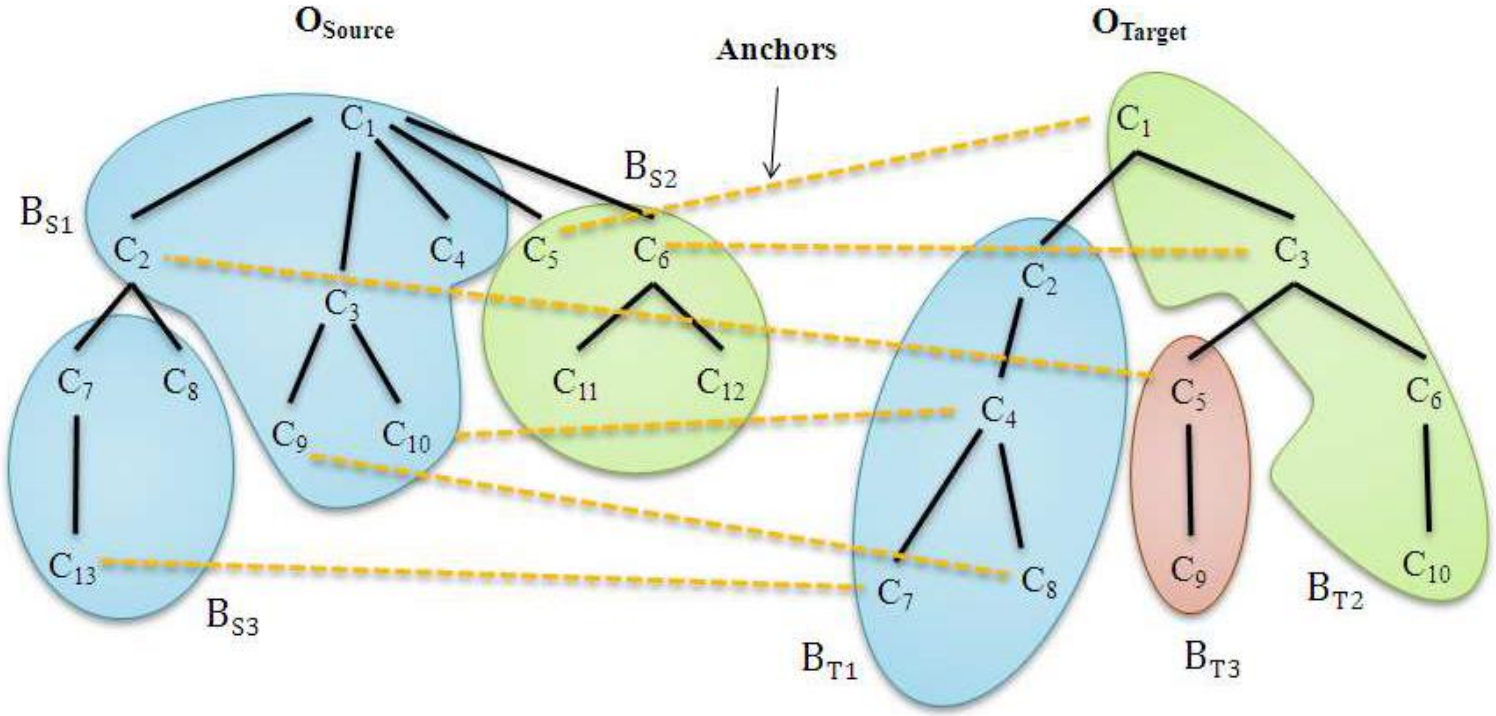
Zwei generelle Ansätze

- Vermeide kartesisches Produkt durch Eliminierung unnötiger Vergleiche im Vorfeld
  - Anwendung eines „weniger“ komplexen Matchers und Ausgrenzung aller Paare mit niedriger Ähnlichkeit von weiteren Berechnungen → gut bei sequentiellen Workflows
- Partition-based Matching (Blocking)
  - „Teile-Herrsche Prinzip“: Teile Ontologien in Partitionen auf und vergleiche nur Partitionen, welche sich stark ähneln
  - Beispiel für zwei Anatomie-Ontologien
    - „nervous system“ ≠ „body fluid or substance“
    - „cardiovascular system“ = „circulatory system“



Partition-based matching (Falcon-AO, Taxomap)

- Strukturelles Clustering der Ontologien für Partitionierung
- Ähnlichkeit von Partitionen über Anker (*anchors*)
  - Anker: sehr stark ähnelnde Konzepte



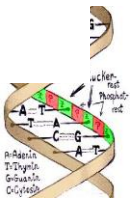
Hamdi, Safar, Reynaud, Zargayouna: Alignment-based partitioning of large-scale ontologies. Advances in Knowledge Discovery and Management, 2009.

Paralleles Ontology Matching

Grundprinzip

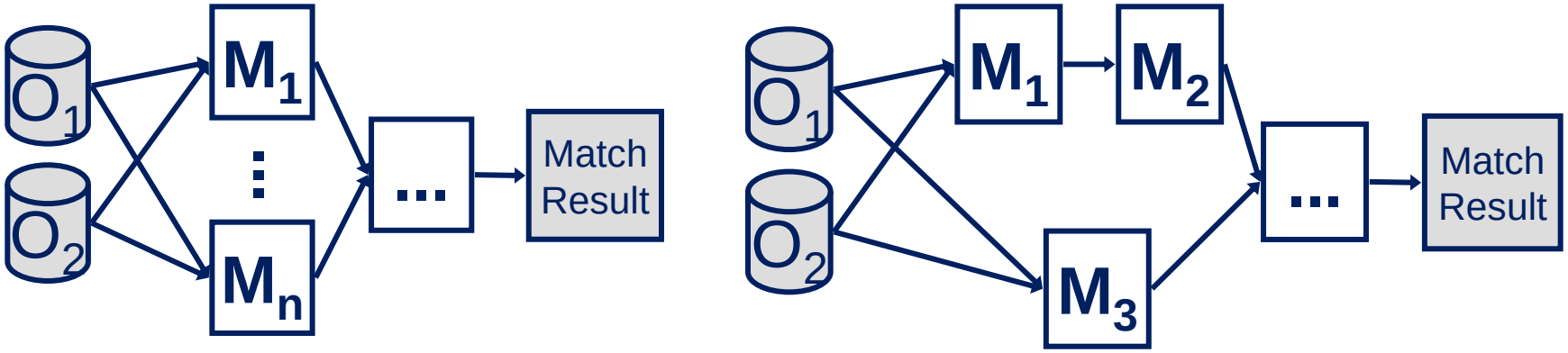
- Verteile Match-Anwendung auf mehrere CPUs / Rechenknoten
  - „Kill it with iron“-Technik
  - Neue Hardware sehr gut Multi-Threading fähig (Dual / Quad Core CPUs) jedoch oft nicht ausgenutzt
- Zwei prinzipielle Möglichkeiten der Verteilung *
  - Verteilung auf Matcher-Ebene (*inter parallelization*)
    - Eine CPU führt jeweils einen Matcher aus
  - Parallelisierung einzelner Matcher (*intra parallelization*)
    - Eine CPU evaluiert lediglich Teil des kartesischen Produkts
  - Kombination aus beiden möglich (*inter-intra*)

** Groß, Hartung, Kirsten, Rahm: On matching large life science ontologies in parallel. Data Integration in the Life Sciences, 2010.*



Inter Matcher Parallelization

Parallele Ausführung unabhängig ausführbarer Matcher

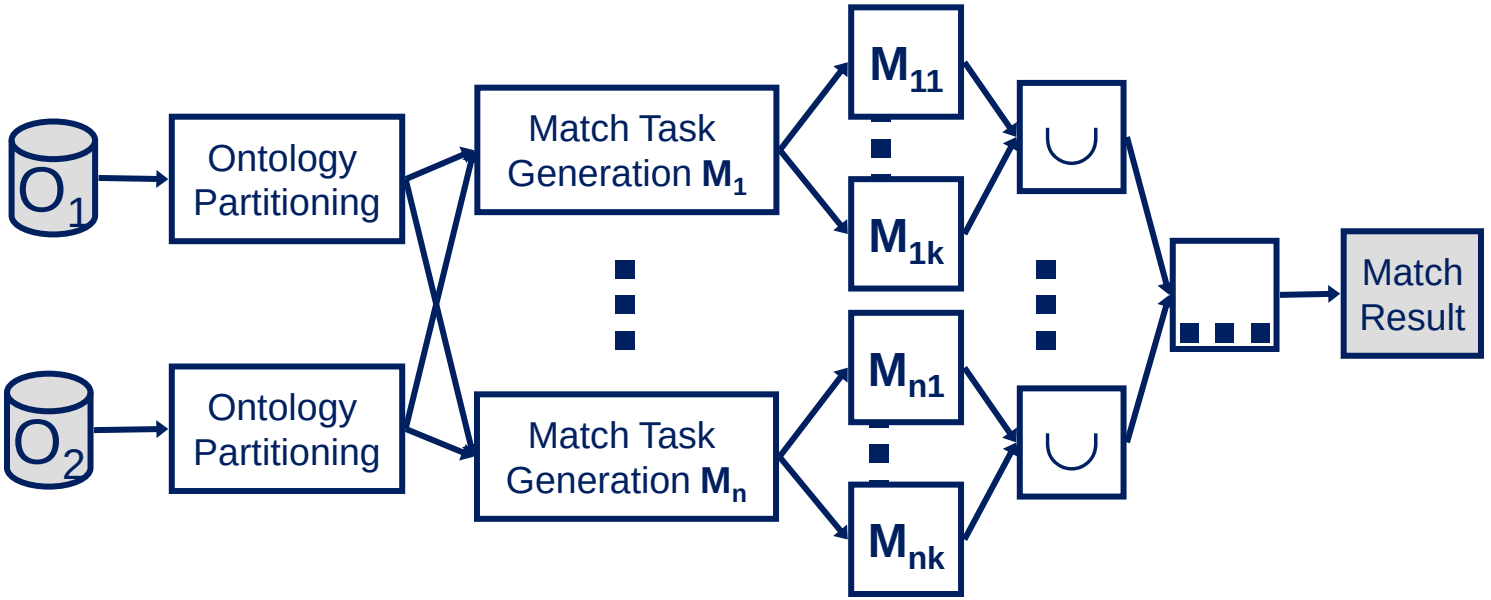


- Kann Laufzeit bis auf das n-fache reduzieren (n = |Matcher|)
- Anzahl parallel ausführbarer Matcher limitiert Parallelisierung
- Langsamster Matcher „bremst“ speed up
- Speicheranforderungen bleiben (keine Datenpartitionierung)

Intra Matcher Parallelization

Generierung von Match-Teilaufgaben zur Ausführung auf mehreren CPUs

- *Basis:* Partitionierung der Ontologien

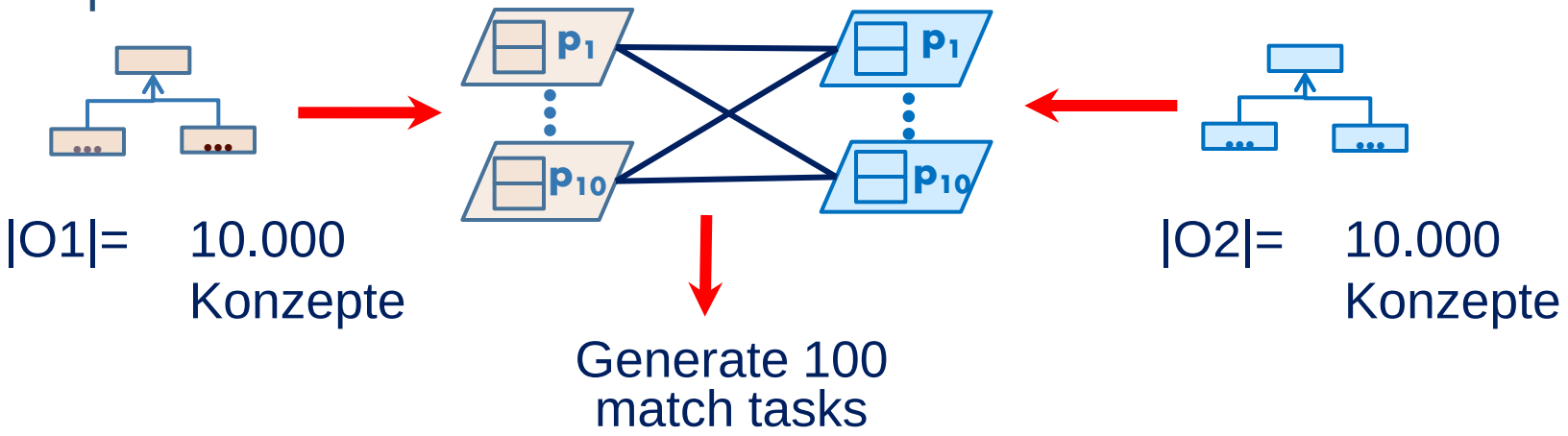


- Reduktion der Speicheranforderungen
- Auch für sequentielle Match-Workflows anwendbar

Partitionierung der Ontologien

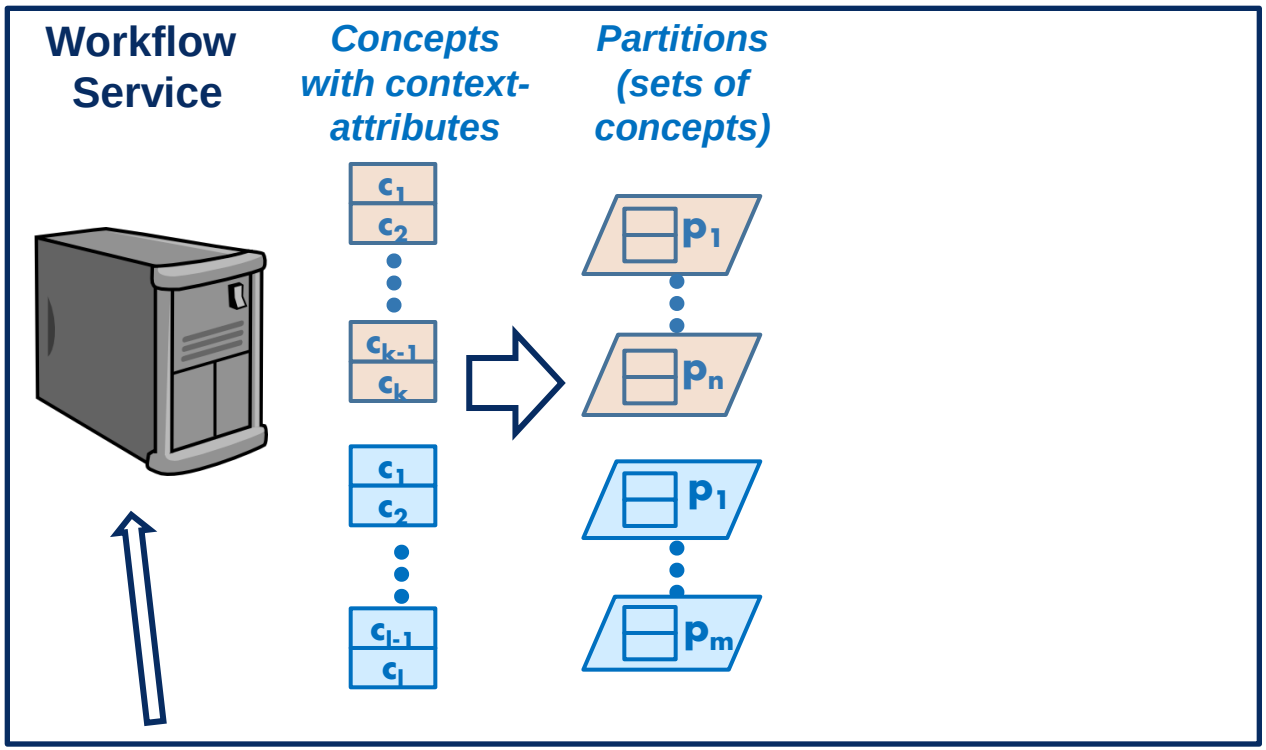
Aufteilung der Ontologien in Partitionen gleicher Größe (Anzahl Konzepte)

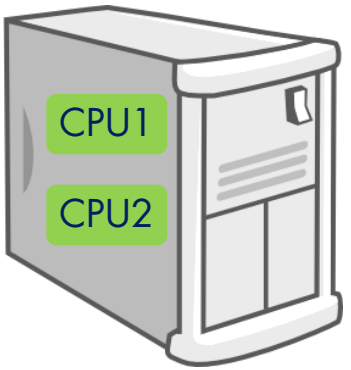
- Jeder Match-Task vergleicht eine Partition aus O1 mit einer Partition aus O2
- Beispiel:



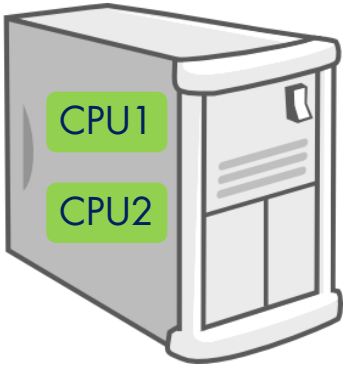
- Für große Ontologien skalierbar (Lastbalancierung)
- Alle Vergleiche werden ausgeführt (→ Match-Qualität)
- Für verschiedene Matcher anwendbar

Infrastruktur und Ausführungsbeispiel

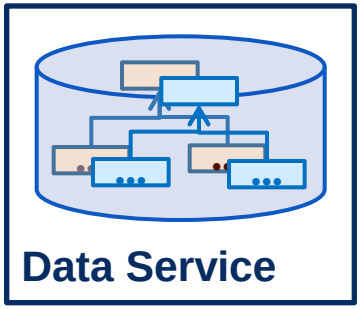


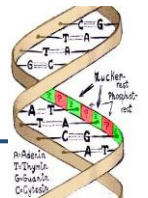


Match Service 1

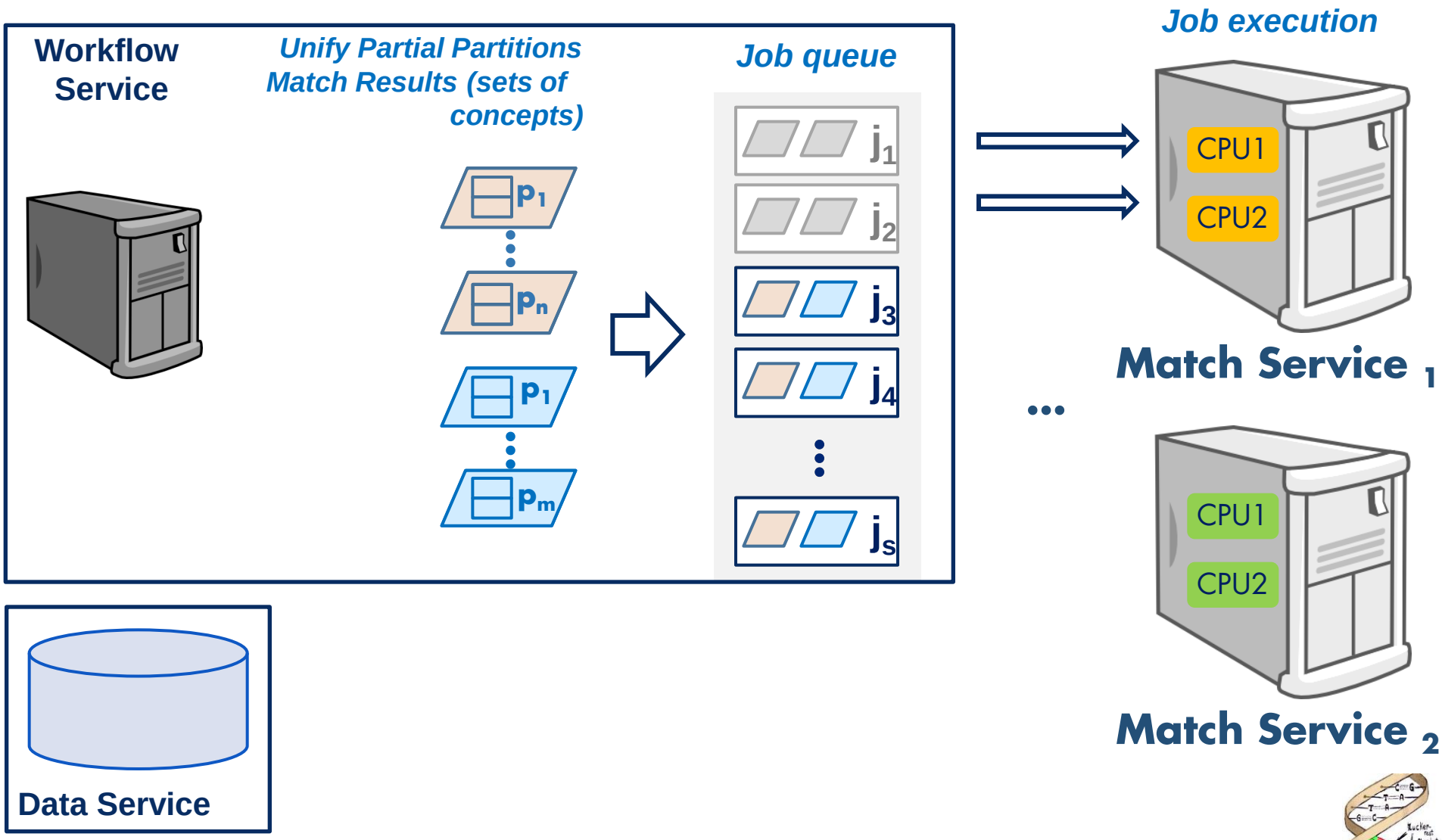


Match Service 2

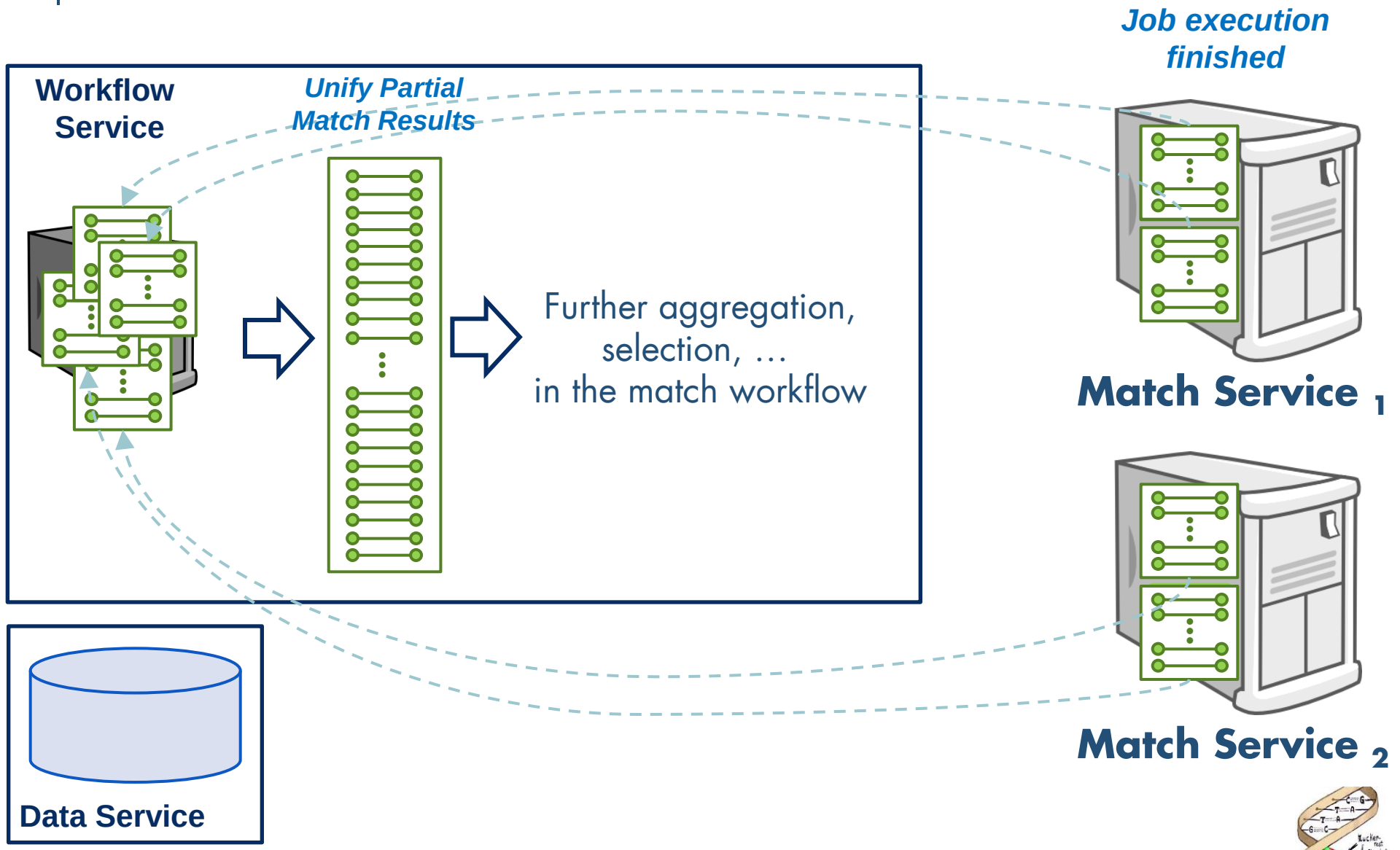




Infrastruktur und Ausführungsbeispiel



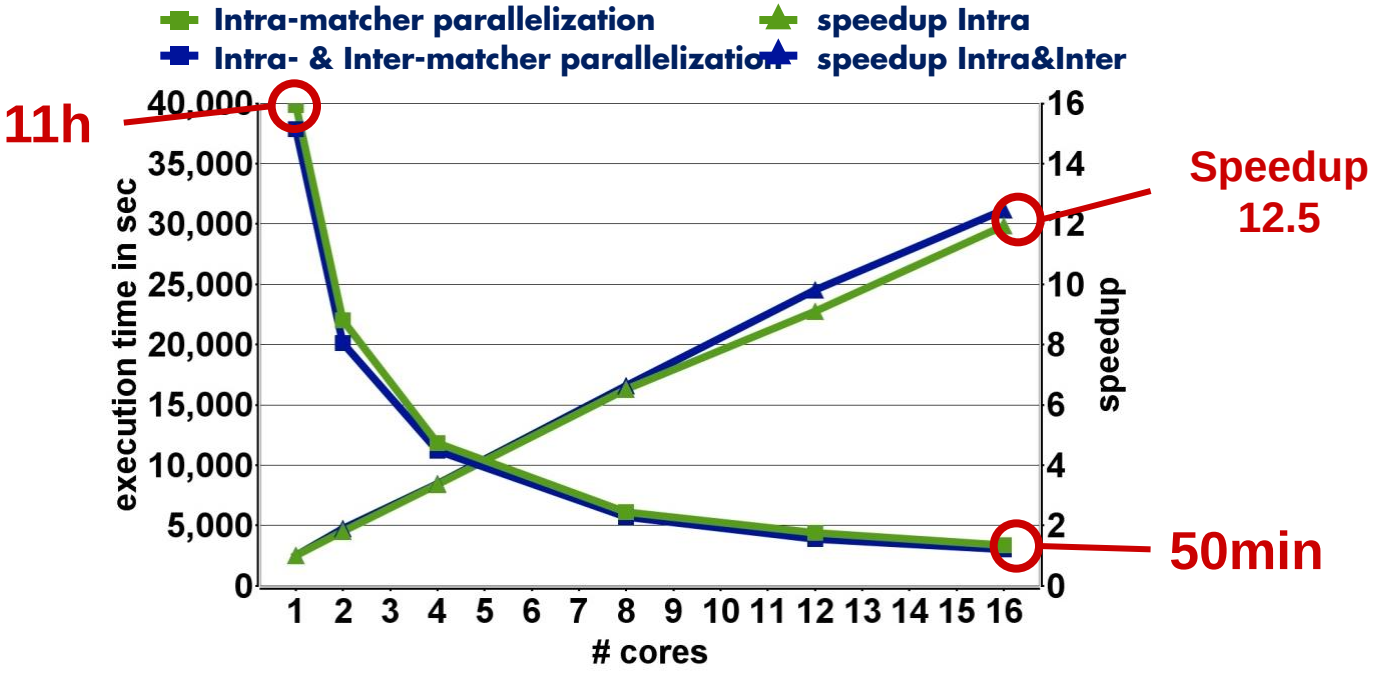
Infrastruktur und Ausführungsbeispiel



Evaluierung

Match-Problem: Molekulare Funktionen (~10.000) vs. Biologische Prozesse (~20.000) der Gene Ontology

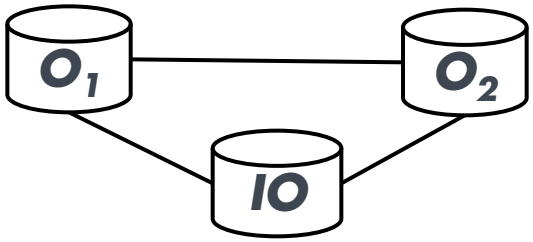
- 4 Rechenknoten mit jeweils 4 CPUs
- Kombination von 3 Matchern: Name/Synonym, NamePath, Children

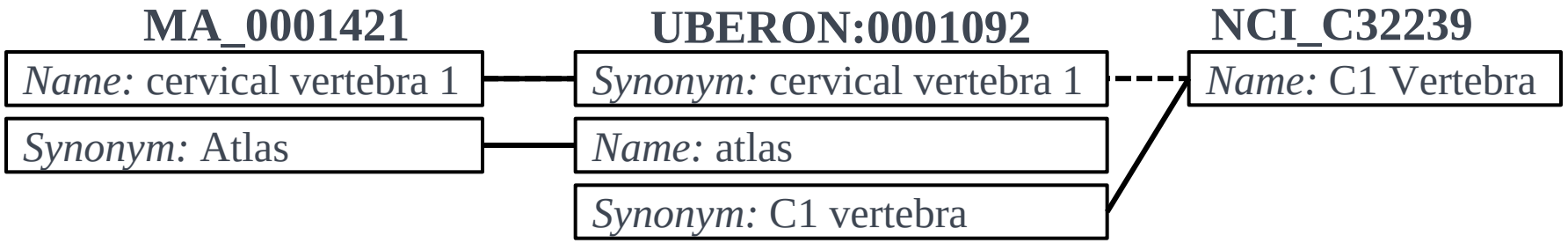


Komposition von Mappings

Indirektes (composition-based) Matching

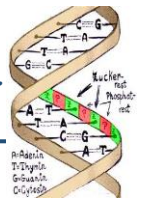
- Ausnutzung bereits existierender Mappings (reuse) zu einer Mediator/Intermediate Ontologie (IO)
  - Hub (wichtige zentrale Terminologie)
  - Synonym-Verzeichnis



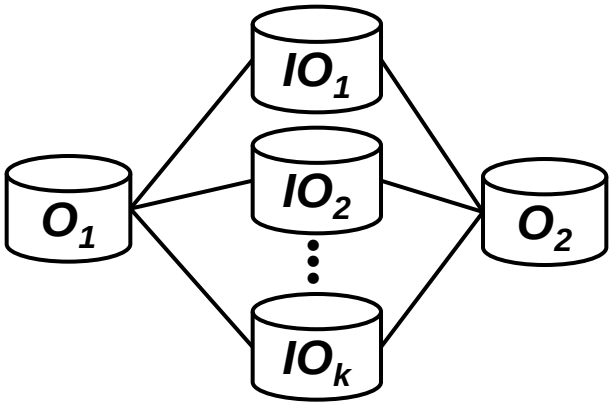


- Vorteile
  - Erkennung neuer Korrespondenzen durch Komposition
  - Einsparung von Rechenaufwand (kein kartesisches Produkt)

Groß, Hartung, Kirsten, Rahm: Mapping Composition for Matching Large Life Science Ontologies. Intl. Conference on Biomedical Ontology, 2011.

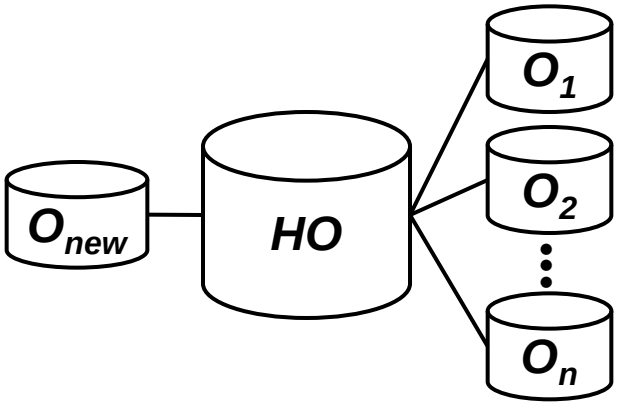


Möglichkeiten von Komposition



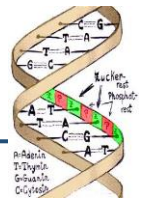
Komposition über mehrere Mediatorontologien

- IOs sollten starke Ähnlichkeit mit O1 und O2 aufweisen
- Gegenseitige Ergänzung ausnutzen



Zentraler Hub vorhanden

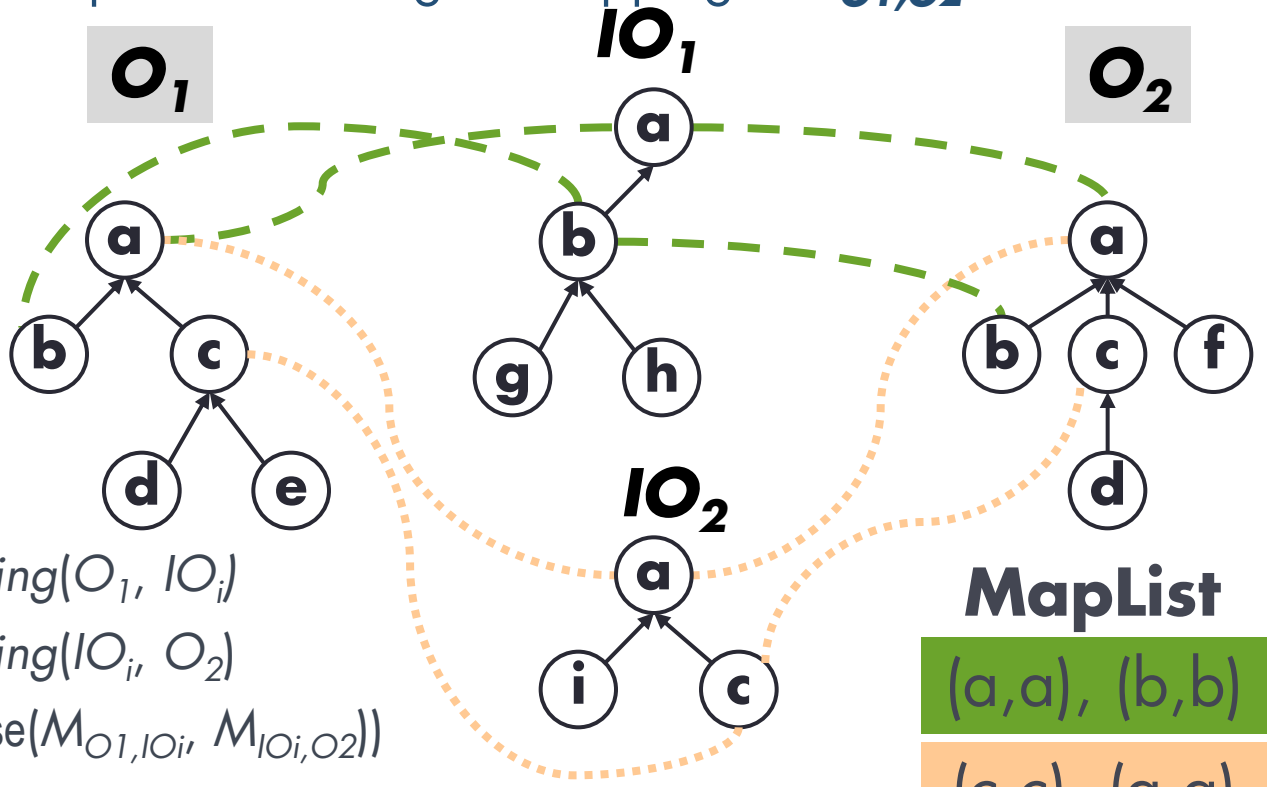
- Hub besitzt bereits zahlreiche Mappings zu anderen Ontologien
- Effizientes Matching einer neuen Ontologie zu allen anderen möglich



ComposeMatch

Eingabe: Ontologien O_1 und O_2, Liste von Mediatoren $IO_1 \ldots IO_k$, minimales Vorkommen *occ*

Ausgabe: Mittels Komposition erzeugtes Mapping $CM_{O1,O2}$



MapList ← empty
for each $IO_i \in IO$ **do**
  $M_{O1,IOi} \leftarrow getMapping(O_1, IO_i)$
  $M_{IOi,O2} \leftarrow getMapping(IO_i, O_2)$
  MapList.add(compose($M_{O1,IOi}$, $M_{IOi,O2}$))
end for
return merge(*MapList, occ*)

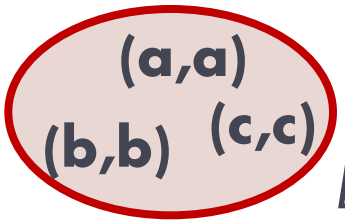
occ = 1: $CM_{O1,O2} = \{(a,a),(b,b),(c,c)\}$
occ = 2: $CM_{O1,O2} = \{(a,a)\}$

ExtendMatch

Eingabe: Ontologien O_1 and O_2, mittels Komp.erzeug.Mapping $CM_{O1,O2}$
Ausgabe: Komplettes Mapping $EM_{O1,O2}$

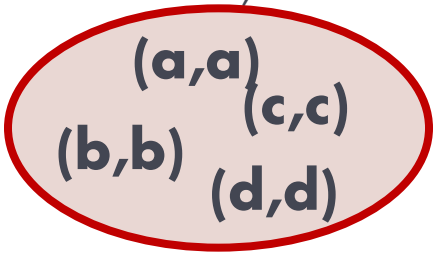
$\Delta O_1 \leftarrow$ extract(O_1, $CM_{O1,O2}$)
$\Delta O_2 \leftarrow$ extract(O_2, inverse($CM_{O1,O2}$))
$DM_{\Delta O1 \Delta O2} \leftarrow$ match(ΔO_1, ΔO_2) *//Herkömmlicher Match*
$EM_{O1,O2} \leftarrow$ merge({$CM_{O1,O2}$, $DM_{\Delta O1 \Delta O2}$}, 1)
return $EM_{O1,O2}$

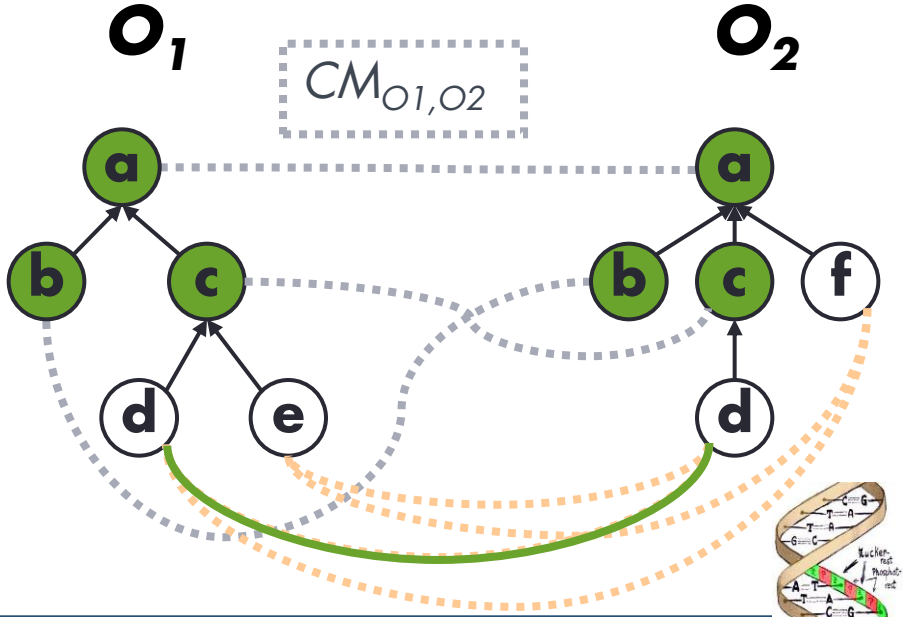
$CM_{O1,O2}$



$DM_{O1,O2}$

$EM_{O1,O2}$

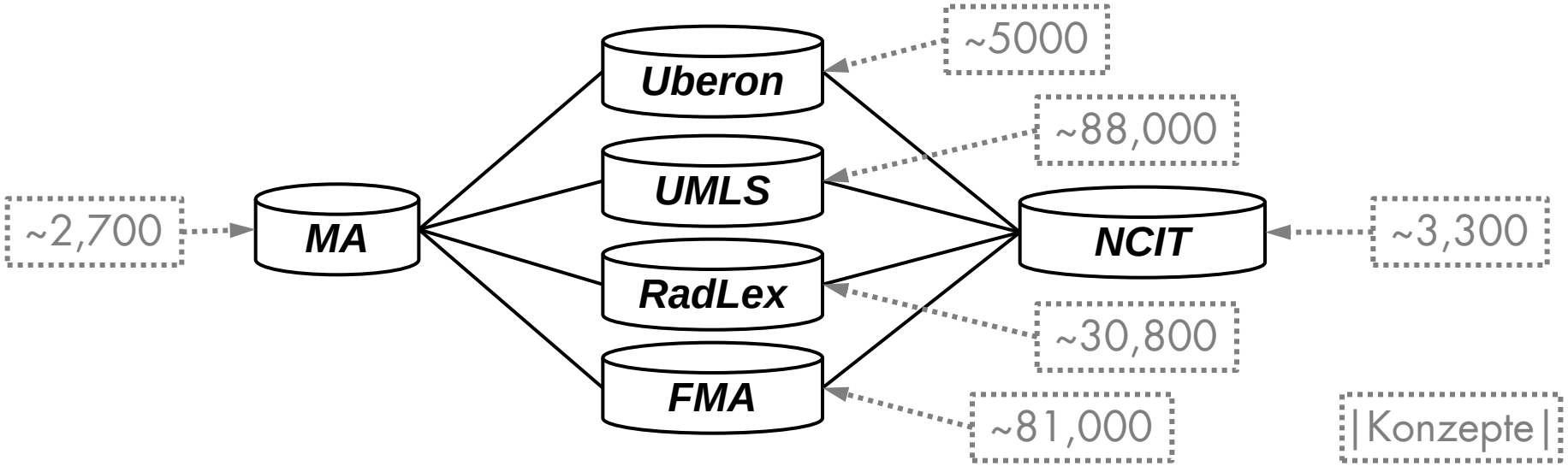




Evaluierung

Anatomie Match-Problem von OAEI

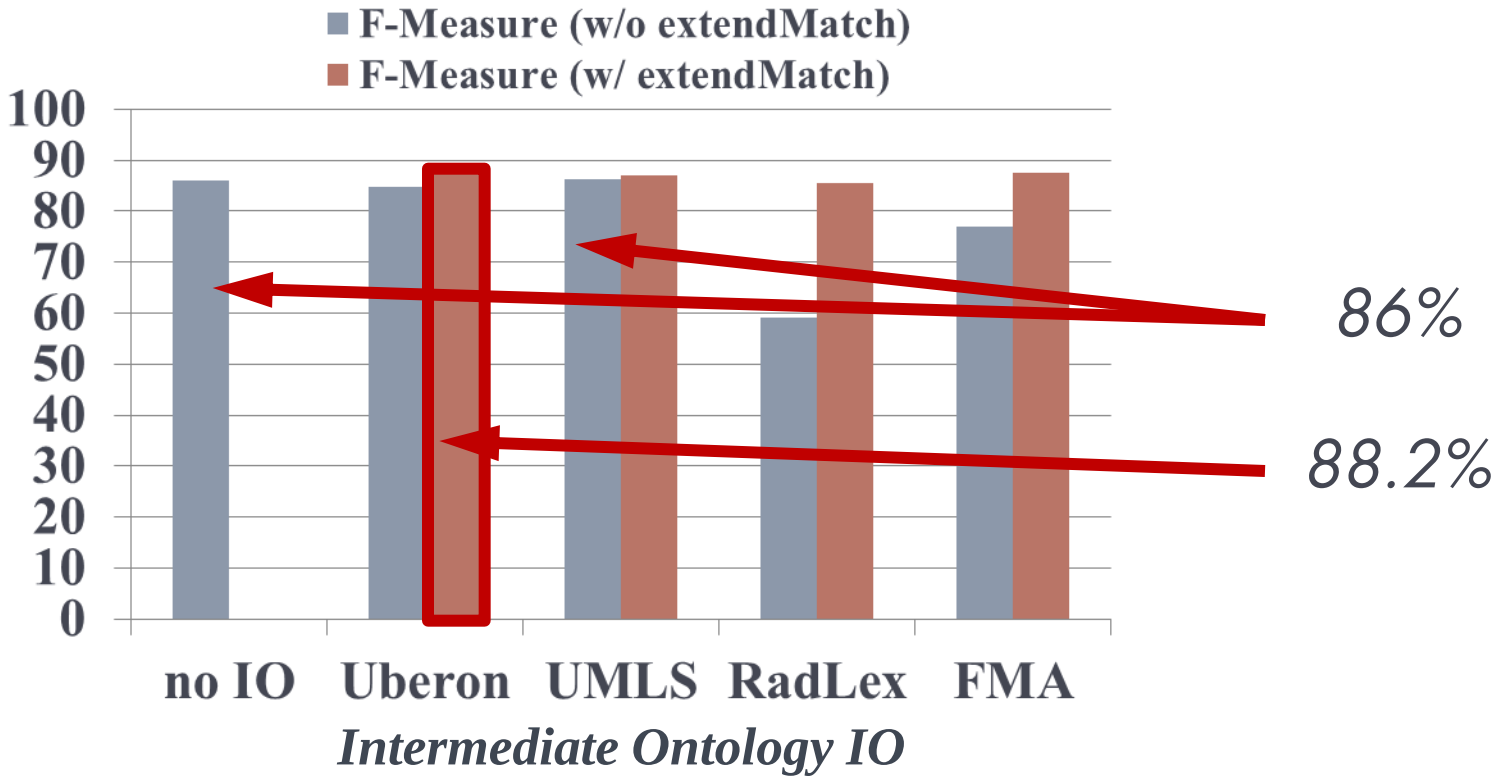
- NCI-Thesaurus (Anatomieteil) - Adult Mouse Anatomy



- Perfektes Mapping: ca. 1.500 Korrespondenzen
- Andere Mappings MA-IO, IO-NCIT vorberechnet (alternativ: Nutzen vorhandener bestätigter Mappings)

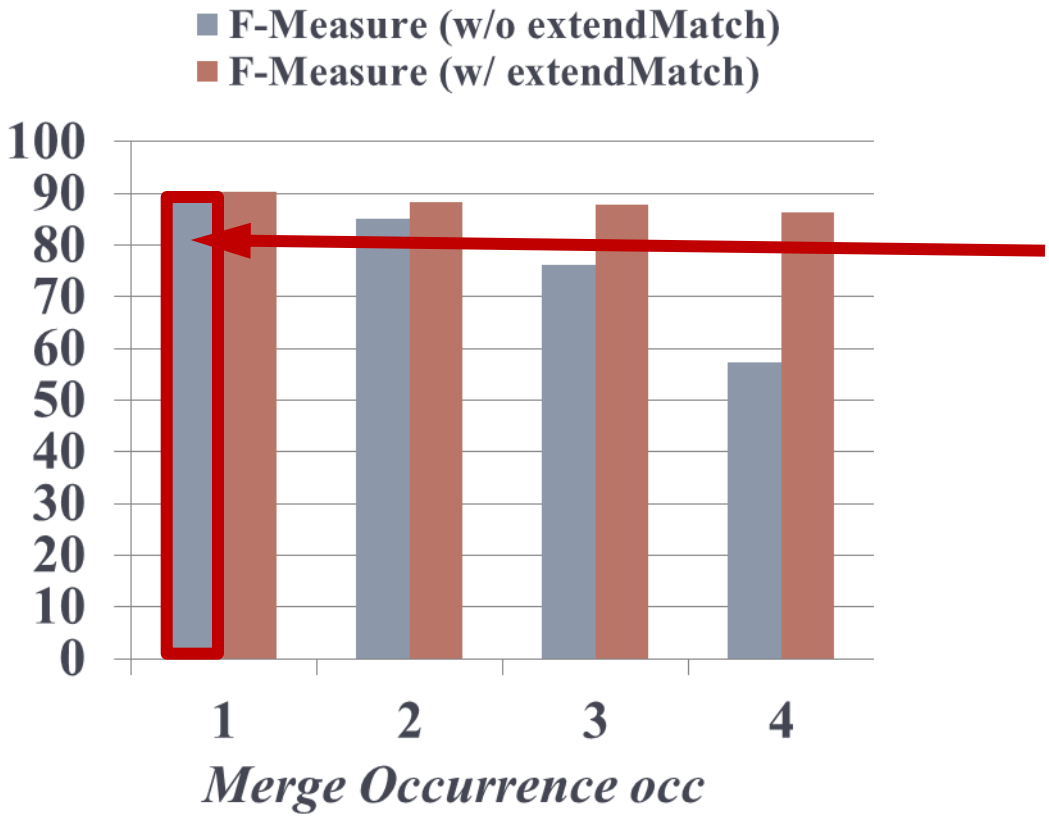
Komposition über einen Mediator

- Vergleich mit direktem Match-Ergebnis
- Zusätzliches Matching der nicht durch *composeMatch* abgedeckten Teile (*extendMatch*)



Komposition über alle Mediatoren

- Testen verschiedener Mehrheitsentscheidungen
  occ = 1, …, 4



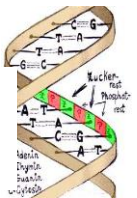
union(occ=1)

F-Measure	**90.2**
Precision	92.7
Recall	87.8

GOMMA @ OAEI 2012 *

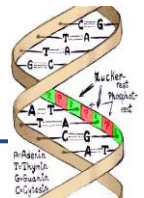
- GOMMA System (Generic Ontology Matching and Mapping Management)
  - Spezialisierung auf das Matchen großer Ontologien
  - Parallelisierung auf mehreren CPU Cores
  - Anwendung von Mapping-Komposition und Blocking
- OAEI 2012
  - 21 Teilnehmer / 53 Tests (Qualität u. Laufzeit als Kriterien)
  - Tracks: Benchmark, Conference, Anatomy, LargeBio, Library, Multifarm
- Hauptergebnisse
  - Alle Tasks erfolgreich absolviert
  - Sieger bzgl. Qualität in Anatomy und Library
  - Sehr gut bzgl. Laufzeit

** Groß, Hartung, Kirsten, Rahm: GOMMA Results for OAEI 2012. Ontology Matching Workshop @ ISWC, 2012.*



Zusammenfassung

- **Ontology Matching**
  - Wichtiger Prozess zur (semi-)automatischen Erstellung von Mappings zwischen Ontologien
    - Anwendungen: Anfrageverarbeitung, Datenintegration, …
  - Vers. Ansätze (element-, structure-, instance-based)
  - Kombination von Ansätzen zur Erhöhung der Match-Qualität
- **Large-Scale Ontology Matching**
  - Reduzierung der Komplexität des Match-Problems
  - Vers. Techniken
    - Pruning, Blocking
    - Parallel Matching
    - Reuse, Komposition



Fragen ?