ConQuR-Bio

Consensus ranking with Query Reformulation for biological data

Bryan Brancotte\textsuperscript{1}, Bastien Rance\textsuperscript{2}, Alain Denise\textsuperscript{1,3}, Sarah Cohen-Boulakia\textsuperscript{1}

(1) Laboratoire de Recherche en Informatique (LRI) & INRIA, Université Paris-Sud, France
(2) Biomedical Informatics and Public Health Department, University Hospital Georges Pompidou, AP-HP, Paris, France & INSERM, Université Paris Descartes, Sorbonne Paris Cité, Faculté de médecine, Paris, France
(3) Institut de Génétique et de Microbiologie (IGM), CNRS UMR 8621 Université Paris-Sud - France

Published in: Data Integration in the Life Sciences (DILS). T. 8574. LNCS. 2014

Journées Ouvertes en Biologie, Informatique & Mathématiques
July 7\textsuperscript{th}, 2014
An approach driven by use cases

Searching for relevant genes for a disease?
Connecting to the NCBI...

An approach driven by use cases

NCBI portal

Cervix cancer
→ 460 genes returned

Cervical cancer
→ 20 genes returned

Equivalent reformulations: cervix cancer vs cervical cancer
An approach driven by use cases

NCBI portal

ADHD
→ 109 genes returned
Attention deficit hyperactivity disorders → 144 genes

Only 74 genes in common!

Equivalent reformulations: cervix cancer vs cervical cancer

Abbreviations: Attention deficit hyperactivity disorders vs ADHD
An approach driven by use cases

**NCBI portal**

- Breast cancer tumor suppressor: → 681 genes
- Breast cancer tumour suppressor: → 291 genes

**Equivalent reformulations**: cervix cancer vs cervical cancer

**Abbreviations**: Attention deficit hyperactivity disorders vs ADHD

**Lexical-based reformulations**: tumor suppressor vs tumour suppressor

Regional settings are also important
An approach driven by use cases

**NCBI portal**

When querying with "Lynch Syndrome" 6 new genes are returned compared to "colorectal cancer"

**Equivalent reformulations:** cervix cancer vs cervical cancer

**Abbreviations:** Attention deficit hyperactivity disorders vs ADHD

**Lexical-based reformulations:** tumor suppressor vs tumour suppressor

**Narrower-term-based reformulations:** The colorectal cancer versus a subtype: the Lynch Syndrome
An approach driven by use cases

**Our goals**

- **Consider reformulations** to have complete sets of answers
- Propose **one list** of results leveraging all the lists of results.
- All in an **on-the-fly** approach.

**Equivalent reformulations**: cervix cancer vs cervical cancer

**Abbreviations**: Attention deficit hyperactivity disorders vs ADHD

**Lexical-based reformulations**: tumor suppressor vs tumour suppressor

**Narrower-term-based reformulations**: The colorectal cancer versus a subtype: the Lynch Syndrome
Use cases

ConQuR-Bio approach

Rank Aggregation Problem

Results

Conclusion
Use cases

ConQuR-Bio approach

Rank Aggregation Problem

Results

Conclusion

Online at http://conqur-bio.lri.fr/
Use cases

ConQuR-Bio approach

Rank Aggregation Problem

Results

Conclusion

Online at http://conqur-bio.lri.fr/
How to address all the problem induced by use cases?

How to obtain a **relevant** list of genes taking into account **all reformulations**?

We have automatized the extraction of:
- query reformulations from different biomedical terminologies (MeSH, OMIM, ICD10CM, ICD9CM, SNOMED CT)
- lists of genes sorted with the "relevance" criteria of the NCBI for each reformulation

We want to:
- take into account **all the genes** returned by each reformulation
- exploit the fact that some genes appear in the results of several reformulations while others don’t
- provide **one single list** of genes answering the user’s query ranked by relevance

⇒ The Rank Aggregation Problem can answer the needs!
How to address all the problem induced by use cases?

How to obtain a relevant list of genes taking into account all reformulations?

We have automatized the extraction of:
- query reformulations from different biomedical terminologies (MeSH, OMIM, ICD10CM, ICD9CM, SNOMED CT)
- lists of genes sorted with the "relevance" criteria of the NCBI for each reformulation

We want to:
- take into account all the genes returned by each reformulation
- exploit the fact that some genes appear in the results of several reformulations while others don’t
- provide one single list of genes answering the user’s query ranked by relevance

⇒ The Rank Aggregation Problem can answer the needs!
How to address all the problem induced by use cases?

How to obtain a relevant list of genes taking into account all reformulations?

We have automatized the extraction of:
- query reformulations from different biomedical terminologies (MeSH, OMIM, ICD10CM, ICD9CM, SNOMED CT)
- lists of genes sorted with the "relevance" criteria of the NCBI for each reformulation

We want to:
- take into account all the genes returned by each reformulation
- exploit the fact that some genes appear in the results of several reformulations while others don’t
- provide one single list of genes answering the user’s query ranked by relevance

⇒ The Rank Aggregation Problem can answer the needs!
1 Use cases

2 ConQuR-Bio approach

3 Rank Aggregation Problem

4 Results

5 Conclusion
A ranking with ties is an ordering of buckets (set of elements) where two elements are ranked differently iff they are in different buckets and are tied otherwise.

\[
\{\{ A \}, \{ C, B \} \}
\]

B and C are tied, both ranked after A

→ to compare rankings with ties we need a distance

The generalized Kendall–\( \tau \) distance [Fag+04a] denoted \( K^{(p)} \)
Counts the pairwise disagreements between two rankings with ties:
- counts 1 when two elements are inversed
- counts \( p \in ]0; 1] \) when two elements are tied in only one ranking

Example: \[
K^{(p)}( \{ \{ C \}, \{ A \}, \{ B \} \}, \{ \{ A \}, \{ C, B \} \} ) = 0_{A-B} + 1_{A-C} + p_{B-C}
\]

→ What is the rank aggregation problem?
A ranking with ties is an ordering of buckets (set of elements) where two elements are ranked differently iff they are in different buckets and are tied otherwise.

→ to compare rankings with ties we need a distance

The generalized Kendall–τ distance [Fag+04a] denoted $K^{(p)}$
Counts the pairwise disagreements between two rankings with ties:
- counts 1 when two elements are **inversed**
- counts $p \in ]0; 1]$ when two elements are **tied** in only one ranking

Example: $K^{(p)}(\{\{C\}, \{A\}, \{B\}\}, \{\{A\}, \{C, B\}\}) = 0_{A-B} + 1_{A-C} + p_{B-C}$

→ What is the rank aggregation problem?
From alternative rankings to one consensus ranking

The Rank Aggregation Problem

Formally, the rank aggregation problem is to find, for a set of input rankings $R$, a consensus ranking $c^*$ such that: $K^{(p)}(R, c^*) \leq K^{(p)}(R, r), \forall r \in \mathcal{R}$;

$K^{(p)}$ is the generalized Kendall$-\tau$ distance [Fag+04a]:
- counts $1$ when two elements are inversed
- counts $p \in ]0; 1]$ when two elements are tied in only one ranking

Example

Let us consider the set of input rankings $R = \{r_1, r_2, r_3\}$, the median under the generalized Kendall$-\tau$ distance is $c$

$$R = \begin{cases} r_1 = [\{A\}, \{D\}, \{B, C\}] \\ r_2 = [\{B\}, \{A\}, \{D\}, \{C\}] \\ r_3 = [\{A, D\}, \{B, C\}] \end{cases}$$

$$c = [\{A\}, \{D\}, \{B, C\}]$$

$$K^{(p)}(R, c) = 1_{A-B \oplus r_2} + 1_{B-D \oplus r_2} + p_{A-D \oplus r_3} + p_{B-C \oplus r_2} = 2 + 2p$$

→ NP-Hard problem[Dwo+01], how to compute a solution?
→ Real case rankings are not over the same elements, how to deal with it?
From alternative rankings to one consensus ranking

The Rank Aggregation Problem

Formally, the rank aggregation problem is to find, for a set of input rankings $R$, a consensus ranking $c^*$ such that: $K^{(p)}(R, c^*) \leq K^{(p)}(R, r), \forall r \in R$;

$K^{(p)}$ is the generalized Kendall-$\tau$ distance [Fag+04a]:
- counts 1 when two elements are inversed
- counts $p \in ]0; 1]$ when two elements are tied in only one ranking

Example

Let us consider the set of input rankings $R = \{r_1, r_2, r_3\}$, the median under the generalized Kendall-$\tau$ distance is $c$

\[
R = \begin{cases}
  r_1 = [{\{A\}, \{D\}, \{B, C\}}] \\
  r_2 = [{\{B\}, \{A\}, \{D\}, \{C\}}] \\
  r_3 = [{\{A, D\}, \{B, C\}}]
\end{cases}
\]

$c = [{\{A\}, \{D\}, \{B, C\}]}$

\[
K^{(p)}(R, c) = 1_{A-B\oplus r_2} + 1_{B-D\oplus r_2} + p_{A-D\oplus r_3} + p_{B-C\oplus r_2} = 2 + 2p
\]

→ NP-Hard problem[Dwo+01], how to compute a solution?
→ Real case rankings are not over the same elements, how to deal with it?
Formally, the rank aggregation problem is to find, for a set of input rankings $R$, a consensus ranking $c^*$ such that: $K^{(p)}(R, c^*) \leq K^{(p)}(R, r)$, $\forall r \in R$;

$K^{(p)}$ is the generalized Kendall-$\tau$ distance [Fag+04a]:
- counts 1 when two elements are inversed
- counts $p \in ]0; 1]$ when two elements are tied in only one ranking

Example

Let us consider the set of input rankings $R = \{r_1, r_2, r_3\}$, the median under the generalized Kendall-$\tau$ distance is $c$

$$R = \begin{cases} r_1 = \{\{ A \}, \{ D \}, \{ B, C \}\} \\ r_2 = \{\{ B \}, \{ A \}, \{ D \}, \{ C \}\} \\ r_3 = \{\{ A, D \}, \{ B, C \}\} \end{cases}$$

$$c = \{\{ A \}, \{ D \}, \{ B, C \}\}$$

$$K^{(p)}(R, c) = 1_{B-D \oplus r_2} + 1_{B-D \oplus r_2} + p_{A-D \oplus r_3} + p_{B-C \oplus r_2} = 2 + 2p$$

→ NP-Hard problem[Dwo+01], how to compute a solution?
→ Real case rankings are not over the same elements, how to deal with it?
From alternative rankings to one consensus ranking
Dealing with missing elements (1/2)

When alternative rankings (results of reformulations) produce different elements, how to apply the distance and construct a complete ranking?

Unification process [CBDH11; SZ09]

Appends missing elements at the end in a unification bucket:

\[
\begin{align*}
    r_1 &= [\{A\}, \{D\}] \\
    r_2 &= [\{B\}, \{A, D\}, \{C\}] \\
\end{align*}
\]

\[
\Rightarrow \begin{align*}
    r_1' &= [\{A\}, \{D\}, \{B, C\}_u] \\
    r_2' &= [\{B\}, \{A, D\}, \{C\}] \\
\end{align*}
\]

\[B\text{ and } C\text{ are indeed less relevant than } A\text{ and } D\text{ in } r_1\]

\[B\text{ and } C\text{ are not equally relevant in } r_1\text{ and should not be considered so in } r_1'!\]
From alternative rankings to one consensus ranking
Dealing with missing elements (1/2)

When alternative rankings (results of reformulations) produce different elements, how to apply the distance and construct a complete ranking?

Unification process [CBDH11; SZ09]

Append missing elements at the end in a unification bucket:

\[
\begin{align*}
    r_1 &= [\{A\}, \{D\}] \\
    r_2 &= [\{B\}, \{A, D\}, \{C\}] \\
\end{align*}
\]

\[
\Rightarrow \begin{cases} 
    r'_1 = [\{A\}, \{D\}, \{B, C\}] \\
    r'_2 = [\{B\}, \{A, D\}, \{C\}] \\
\end{cases}
\]

\[B \text{ and } C \text{ are indeed less relevant than } A \text{ and } D \text{ in } r_1\]

\[B \text{ and } C \text{ are not equally relevant in } r_1 \text{ and should not be considered so in } r'_1!\]
From alternative rankings to one consensus ranking
Dealing with missing elements (1/2)

When alternative rankings (results of reformulations) produce different elements, how to apply the distance and construct a complete ranking?

Unification process [CBDH11; SZ09]

**Append**s missing elements at the end in a **unification bucket**:

\[
\begin{align*}
    & r_1 = [\{A\}, \{D\}] \\
    & r_2 = [\{B\}, \{A, D\}, \{C\}] \\
\end{align*}
\]

\[
\Rightarrow \quad \begin{align*}
    & r'_1 = [\{A\}, \{D\}, \{B, C\}_u] \\
    & r'_2 = [\{B\}, \{A, D\}, \{C\}] \\
\end{align*}
\]

*B and C are indeed less relevant* than *A and D in *r*₁

*B and C are not equally relevant* in *r*₁ and should not be considered so in *r'₁*!
From alternative rankings to one consensus ranking
Dealing with missing elements (2/2)

Extending the generalized Kendall-$\tau$ distance to a pseudometrics $\mathcal{M}(r'_1, r'_2)$

Counts the pairwise disagreements between two rankings with ties:
- counts 1 when two elements are inversed
- counts $p \in ]0; 1]$ when two elements are tied in only one ranking

Example

Let us consider the set of input rankings $R = \{r_1, r_2, r_3\}$, the median under the generalized Kendall-$\tau$ distance is $c$ and the median under the pseudometrics is $c'$

$$R = \begin{cases} r_1 = [\{A\}, \{D\}, \{B, C\}_u] \\ r_2 = [\{B\}, \{A\}, \{D\}, \{C\}] \\ r_3 = [\{A, D\}, \{B, C\}_u] \end{cases} \quad c = [\{A\}, \{D\}, \{B, C\}]$$

$$c' = [\{A\}, \{D\}, \{B\}, \{C\}]$$

$\mathcal{M}(R, c) = 2 + p > \mathcal{M}(R, c') = 2$

How can we actually compute a consensus?
From alternative rankings to one consensus ranking
Dealing with missing elements (2/2)

Extending the generalized Kendall–τ distance to a pseudometrics \( M(r'_1, r'_2) \)

Counts the pairwise disagreements between two rankings with ties:
- counts 1 when two elements are inversed
- counts 0 when two elements are tied in at least one unifying bucket
- counts \( p \in ]0; 1] \) when two elements are tied in only one ranking

Example

Let us consider the set of input rankings \( R = \{ r_1, r_2, r_3 \} \), the median under the generalized Kendall-τ distance is \( c \) and the median under the pseudometrics is \( c' \)

\[
R = \begin{cases} 
    r_1 = \{ \{A\}, \{D\}, \{B, C\}_u \} \\
    r_2 = \{ \{B\}, \{A\}, \{D\}, \{C\} \} \\
    r_3 = \{ \{A, D\}, \{B, C\}_u \} 
\end{cases} \\
\text{c} = \{ \{A\}, \{D\}, \{B, C\} \} \\
\text{c'} = \{ \{A\}, \{D\}, \{B\}, \{C\} \} \\
M(R, c) = 2 + p > M(R, c') = 2
\]

How can we actually compute a consensus?
From alternative rankings to one consensus ranking

Computing a solution

Plethora of algorithms [ACN08; Ail10; Bor81; Fag+04b; FKS03b; Mei+07]. Based on our work [Bra+15]a, BioConsert is the algorithm to use.


Main algorithm used: a greedy local search algorithm, BioConsert [CBDH11]: Starting with each input rankings and considering two edit operations (moving an element to a new bucket or an existing bucket), it applies operators as long as it reduces the distance between the current consensus obtained and the input rankings.

\[ \{A\},\{B\},\{C,D\} \rightarrow \{A\},\{D\},\{B\},\{C\} \rightarrow \{A\},\{D\},\{B,C\} \]

Figure: edit operations in BioConsert

Tuning BioConsert: Starting from the best solution provided by BordaCount [Bor81], MEDRank [FKS03a], and Ailon’s 2-approximation [Ail10]

Up to a hundred times faster!
From alternative rankings to one consensus ranking
Computing a solution

Plethora of algorithms [ACN08; Ail10; Bor81; Fag+04b; FKS03b; Mei+07]. Based on our work [Bra+15], BioConsert is the algorithm to use. 

\[\text{Main algorithm used: a greedy local search algorithm, BioConsert [CBDH11]:}\]

Starting with each input rankings and considering two edit operations (moving an element to a new bucket or an existing bucket), it applies operators as long as it reduces the distance between the current consensus obtained and the input rankings.

\[\text{Figure: edit operations in BioConsert}\]

Tuning BioConsert: Starting from the best solution provided by BordaCount [Bor81], MEDRank [FKS03a], and Ailon’s 2-approximation [Ail10]

Up to a hundred times faster!
Plethora of algorithms [ACN08; Ail10; Bor81; Fag+04b; FKS03b; Mei+07]. Based on our work [Bra+15]a, BioConsert is the algorithm to use.


Main algorithm used: a greedy local search algorithm, BioConsert [CBDH11]: Starting with each input rankings and considering two edit operations (moving an element to a new bucket or an existing bucket), it applies operators as long as it reduces the distance between the current consensus obtained and the input rankings.

Figure: edit operations in BioConsert

Tuning BioConsert: Starting from the best solution provided by BordaCount [Bor81], MEDRank [FKS03a], and Ailon’s 2-approximation [Ail10]

Up to a hundred times faster!
1 Use cases

2 ConQuR-Bio approach

3 Rank Aggregation Problem

4 Results

5 Conclusion
Bibliometrics indicators

Focusing on the top 20 genes. For each gene we consider publications co-citing the gene name and the query, and compare to the top 20 genes returned by the NCBI portal.

Using the number of publications
Summing the number of publications co-citing the gene name and the query. With ConQuR-Bio: 56% more publications associated than with the NCBI.

Using publication "freshness"
Averaging the number of days since the last publication co-citing the gene name and the query. With ConQuR-Bio: 25% less days than with the results provided by the NCBI.
Gold-standards

Clinicians of the Institut Curie (France) and the Children’s Hospital of Philadelphia (PA, USA) provided gold-standards, list of expected genes, for 9 different diseases: 7 cancers (bladder, breast, cervical, colorectal, neuroblastoma, prostate, retinoblastoma), one heart disease (the Long QT Syndrome), and one psychiatric disorder (the attention deficit (with) hyperactivity disorder). Diseases are often combined with terms tumor suppressor and oncogene.

Measure: the AUC

The Area Under the ROC Curve [Bra97] is closely related to precision and recall measures, and allows to highlight the presence of elements of the gold standard in the top results. It provides a number in [0, 1], 1 being the highest score.

Examples with two expected results (••)

AUC(•••••)=0.50
AUC(•••••)=0.66
AUC(••••••)=0.66
AUC(•••••••)=0.83
Expertise based indicator

Gold-standards

Clinicians of the Institut Curie (France) and the Children’s Hospital of Philadelphia (PA, USA) provided gold-standards, list of expected genes, for 9 different diseases: 7 cancers (bladder, breast, cervical, colorectal, neuroblastoma, prostate, retinoblastoma), one heart disease (the Long QT Syndrome), and one psychiatric disorder (the attention deficit (with) hyperactivity disorder). Diseases are often combined with terms tumor suppressor and oncogene.

Measure: the AUC

The Area Under the ROC Curve [Bra97] is closely related to precision and recall measures, and allows to highlight the presence of elements of the gold standard in the top results. It provides a number in [0, 1], 1 being the highest score.

Examples with two expected results (••)

- AUC(•••••)=0.50
- AUC(•••••)=0.66
- AUC(•••••)=0.66
- AUC(•••••)=0.83
**Results:** using expertise of clinician collaborators

**Figure:** AUC for the 20 first genes returned by ConQuR-Bio and the NCBI WebSearch for single-term queries.

Average AUC increased of 58%
Results: using expertise of clinician collaborators

Figure: AUC for the 20 first genes returned by ConQuR-Bio and the NCBI WebSearch for (a) Single-term queries, (b) lexical variation around *cervix cancer tumor suppressor*, and (c) the remaining query for which we have gold standards.

Average AUC of ConQuR-Bio results are increased of 44%
1. Use cases

2. ConQuR-Bio approach

3. Rank Aggregation Problem

4. Results

5. Conclusion
ConQuR-Bio...:
- exploits biomedical terminologies to reformulate the user query
- proposes a consensus emphasizing agreements between reformulations results
  - introducing a new pseudometrics answering problematics brought by the data
  - leveraging state-of-the-art algorithms to efficiently propose a relevant consensus
- increases the results AUC of +44% compared to the NCBI
- follows an on-the-fly approach
- is free to use at http://conqur-bio.lri.fr

Ongoing work:
- Consider fine grain recognition of terms in users queries.
- Extend to larger and customizable amounts of biomedical terminologies.


Marina Meila, Kapil Phadnis, Arthur Patterson, and Jeff Bilmes. “Consensus ranking under the exponential model”. In: (2007).

From alternative rankings to one consensus ranking
Notions for the Rank Aggregation Problem

**A ranking with ties** is an ordering of buckets (set of elements) where two elements are ranked differently iff they are in different buckets and are tied otherwise.

→ to compare rankings with ties we need a distance

The generalized Kendall–τ distance [Fag+04a] denoted \( K^{(p)} \)
Counts the pairwise disagreements between two rankings with ties:
- counts 1 when two elements are inverted
- counts \( p \in ]0; 1] \) when two elements are tied in only one ranking

Example: \( K^{(p)}(\left\{ \{C\}, \{A\}, \{B\} \right\}, \left\{ \{A\}, \{C, B\} \right\}) = 0_{A–B} + 1_{A–C} + p_{B–C} \)

→ What is the rank aggregation problem?
Formally, the rank aggregation problem is to find, for a set of input rankings $R$, a consensus ranking $c^*$ such that: $K(p)(R, c^*) \leq K(p)(R, r), \forall r \in \mathcal{R}$;

$K(p)$ is the generalized Kendall-$\tau$ distance [Fag+04a]:
- counts 1 when two elements are inversed
- counts $p \in ]0; 1]$ when two elements are tied in only one ranking

Example

Let us consider the set of input rankings $R = \{r_1, r_2, r_3\}$, the median under the generalized Kendall-$\tau$ distance is $c$

$$R = \begin{cases} r_1 = [\{A\}, \{D\}, \{B, C\}] \\ r_2 = [\{B\}, \{A\}, \{D\}, \{C\}] \\ r_3 = [\{A, D\}, \{B, C\}] \end{cases}$$

$$c = [\{A\}, \{D\}, \{B, C\}]$$

$$K(p)(R, c) = 1_{A-B@r_2} + 1_{B-D@r_2} + p_{A-D@r_3} + p_{B-C@r_2} = 2 + 2p$$

→ NP-Hard problem[Dwo+01], how to compute a solution?
→ Real case rankings are not over the same elements, how to deal with it?
Formally, the rank aggregation problem is to find, for a set of input rankings $R$, a consensus ranking $c^*$ such that: $K^{(p)}(R, c^*) \leq K^{(p)}(R, r), \forall r \in \mathcal{R}$;

$K^{(p)}$ is the generalized Kendall-$\tau$ distance [Fag+04a]:

- counts 1 when two elements are inversed
- counts $p \in ]0; 1]$ when two elements are tied in only one ranking

Example

Let us consider the set of input rankings $R = \{r_1, r_2, r_3\}$, the median under the generalized Kendall-$\tau$ distance is $c$

$$R = \begin{cases}
  r_1 = [\{A\}, \{D\}, \{B, C\}] \\
  r_2 = [\{B\}, \{A\}, \{D\}, \{C\}] \\
  r_3 = [\{A, D\}, \{B, C\}]
\end{cases}$$

$$c = [\{A\}, \{D\}, \{B, C\}]$$

$$K^{(p)}(R, c) = 1_{A-B@r_2} + 1_{B-D@r_2} + p_{A-D@r_3} + p_{B-C@r_2} = 2 + 2p$$

→ NP-Hard problem[Dwo+01], how to compute a solution?
→ Real case rankings are not over the same elements, how to deal with it?
From alternative rankings to one consensus ranking
Dealing with missing elements (1/2)

When alternative rankings (results of reformulations) produce different elements, how to apply the distance and construct a complete ranking?

**Induced Kendall-\(\tau\) distance [Dwo+01]**  
**KO**  
Ignores disagreements related to missing elements

**Projection process [BBN13]**  
**KO**  
Considers only elements appearing in all rankings (it removes the others)

**Unification process [CBDH11; SZ09]**  
OK, but adaptation to do

**Append** missing elements at the end in a **unification bucket**:

\[
\begin{align*}
\{ r_1 &= \left[ \{ A \}, \{ D \} \right] \} &\Rightarrow & \left\{ r'_1 = \left[ \{ A \}, \{ D \}, \{ B, C \} \right] \right\} \\
\{ r_2 &= \left[ \{ B \}, \{ A, D \}, \{ C \} \right] \} &\Rightarrow & \left\{ r'_2 = \left[ \{ B \}, \{ A, D \}, \{ C \} \right] \right\}.
\end{align*}
\]

\( B \) and \( C \) are indeed **less relevant** than \( A \) and \( D \) in \( r_1 \)

\( B \) and \( C \) are **not equally relevant** in \( r_1 \) and should not be considered so in \( r'_1 \)!
From alternative rankings to one consensus ranking
Dealing with missing elements (2/2)

Extending the generalized Kendall–τ distance to a pseudometrics \( \mathcal{M}(r_1', r_2') \)

Counts the pairwise disagreements between two rankings with ties:
- counts 1 when two elements are inversed
- counts \( p \in ]0; 1] \) when two elements are tied in only one ranking

Example

Let us consider the set of input rankings \( R = \{r_1, r_2, r_3\} \), the median under the generalized Kendall-τ distance is \( c \) and the median under the pseudometrics is \( c' \)

\[
R = \begin{cases} 
  r_1 = [\{A\}, \{D\}, \{B, C\}_u] \\
  r_2 = [\{B\}, \{A\}, \{D\}, \{C\}] \\
  r_3 = [\{A, D\}, \{B, C\}_u] 
\end{cases}
\]

\[
c = \begin{bmatrix} \{A\}, \{D\}, \{B, C\} \end{bmatrix}
\]

\[
c' = \begin{bmatrix} \{A\}, \{D\}, \{B\}, \{C\} \end{bmatrix}
\]

\[
\mathcal{M}(R, c) = 2 + p > \mathcal{M}(R, c') = 2
\]

How can we actually compute a consensus?
From alternative rankings to one consensus ranking
Dealing with missing elements (2/2)

Extending the generalized Kendall–τ distance to a pseudometrics $\mathcal{M}(r'_1, r'_2)$

Counts the pairwise disagreements between two rankings with ties:
- counts 1 when two elements are inversed
- counts 0 when two elements are tied in at least one unifying bucket
- counts $p \in ]0; 1]$ when two elements are tied in only one ranking

Example

Let us consider the set of input rankings $R = \{r_1, r_2, r_3\}$, the median under the generalized Kendall-τ distance is $c$ and the median under the pseudometrics is $c'$

$$R = \begin{cases} r_1 = [\{A\}, \{D\}, \{B, C\}_u] \\ r_2 = [\{B\}, \{A\}, \{D\}, \{C\}] \\ r_3 = [\{A, D\}, \{B, C\}_u] \end{cases} \quad c = [\{A\}, \{D\}, \{B, C\}] \\ c' = [\{A\}, \{D\}, \{B\}, \{C\}] \quad \mathcal{M}(R, c) = 2 + p > \mathcal{M}(R, c') = 2$$

How can we actually compute a consensus?
From alternative rankings to one consensus ranking
Computing a solution

Plethora of algorithms [ACN08; Ail10; Bor81; Fag+04b; FKS03b; Mei+07]. Based on our work [Bra+15], BioConsert is the algorithm to use.

\[\textit{Brancotte et al.} \ "\textit{Rank Aggregation with ties: Experiments and Analysis}". \textit{In: Proceedings of the VLDB Endowment} 8.11(2015)\]

Main algorithm used: a greedy local search algorithm, BioConsert [CBDH11]: Starting with each input rankings and considering two edit operations (moving an element to a new bucket or an existing bucket), it applies operators as long as it reduces the distance between the current consensus obtained and the input rankings.

\[
\begin{align*}
\{A\},\{B\},\{C,D\} & \quad \longrightarrow \quad \{A\},\{D\},\{B\},\{C\} & \quad \longrightarrow \quad \{A\},\{D\},\{B,C\} \\
\end{align*}
\]

\textbf{Figure: edit operations in BioConsert}

\textbf{Tuning BioConsert:} Starting from the best solution provided by BordaCount [Bor81], MEDRank [FKS03a], and Ailon’s 2-approximation [Ail10]

\textbf{Up to a hundred times faster!}
Generalized Kendall-$\tau$ distance

Definition

Let $r$ and $c$ be two ranking with ties over $n$ elements. Let $r[i]$ be the rank of $i$ in ranking $r$. The generalized Kendall-$\tau$ distance is:

$$K^{(p)}(r, c) = \#\{(i, j) : r[i] < r[j] \text{ and } c[i] > c[j] \text{ or }$$

$$r[i] > r[j] \text{ and } c[i] < c[j]\} + p \times \#\{(i, j) : r[i] \neq r[j] \text{ and } c[i] = c[j] \text{ or }$$

$$r[i] = r[j] \text{ and } c[i] \neq c[j]\} \quad \text{where } 0 < p \leq 1$$

Distance of a ranking/consensus to a set of rankings

$$K^{(p)} \left( c = \{\{A\}, \{C\}, \{B\}\}, \quad r_1 = \{\{A\}, \{B, C\}\}, \quad r_2 = \{\{C\}, \{A\}, \{B\}\} \right) = 1_{A-C} + p_{B-C}$$
Finding reformulation

For each term, get synonyms from terminologies integrated in the UMLS system.

Two modes

1. Synonym terms: "Breast cancer" and "Malignant neoplasm of breast"

2. Synonym and Narrower terms: *Romano-Ward syndrome* is also known as *lonq qt syndrome type 1*, a subtype of *long qt syndrome*
Reformulation module

Genes for the query "Breast cancer oncogene"

Key-phrase splitting
Find the largest recognized MeSH terms in the provided query

"Breast cancer" reformulations are:
Breast cancer, Breast Carcinoma, Mammary Cancer, Malignant neoplasm of breast, Malignant tumor of breast, ...
Reformulation module

Genes for the query
"Breast cancer oncogene"

Terms found
"Breast cancer" "oncogene"

Key-phrase splitting
Find the largest recognized MeSH terms in the provided query

"Breast cancer" reformulations are:
Breast cancer, Breast Carcinoma, Mammary Cancer, Malignant neoplasm of breast, Malignant tumor of breast, ...
Reformulation module

Genes for the query
"Breast cancer oncogene"

Terms found
"Breast cancer" "oncogene"

Key-phrase splitting
Find the largest recognized MeSH terms in the provided query

"Breast cancer" reformulations are:
Breast cancer, Breast Carcinoma, Mammary Cancer, Malignant neoplasm of breast, Malignant tumor of breast, ...