Supplementary Material (Online Resource) for:
Rank-Based Relational Methods for Interpretable Biomedical Modeling: A Taxonomic Review

Marcin Czajkowski, Krzysztof Jurczuk, and Marek Kretowski

S1 - Taxonomic Glossary of Rank-Based Relational Methods

Supplementary Table 1 provides a broad glossary of rank-based relational methods discussed in the main review. For each method, we list the acronym, full name, primary reference, and a concise classification according to the first two layers of the taxonomy introduced in the main manuscript: Representation and Computation. Where relevant, the third layer, Implementation/Execution, is also noted. This glossary is intended as a quick reference guide; qualitative comparisons are provided in Section S6.

Supplementary Table 1: Taxonomic glossary of rank-based relational methods discussed in the review. Acronyms, primary references, and concise taxonomy assignments are provided for quick reference; qualitative comparisons are reported in Section S6.

Acronym	**Full Name & Reference**	**Taxonomic Classification**
ARXA	Advanced Relative Expression Analysis Czajkowski et al. (2020c)	A top-down induced decision tree using a novel ratio-based scoring method. (Hierarchical Representation; Heuristic search; GPU-accelerated Implementation). See Table 37 for details.
AUCTSP	AUC-based Top Scoring Pair Kagaris et al. (2018)	A single-pair classifier using the AUC statistic as an alternative scoring relation. (Horizontal Representation; Exhaustive Search). See Table 27 for details.
BTSP	Bayesian Top-Scoring Pairs Arslan and Braga-Neto (2017)	A hybrid model integrating a Bayesian Bradley-Terry model with TSP logic. (Embedded Representation; Heuristic Computation). See Table 43 for details.
CFC-CM	Constructing Feature Combinations and Classification Models Lin et al. (2019)	An extension of k-TSP that iteratively fuses discriminative pairs into higher-level features. (Horizontal Representation; Heuristic Search).
Chi-TSG	Chi-square Statistic-based Top-Scoring Genes Wang et al. (2013)	A method using the chi-squared statistic as an alternative relation to select informative gene sets. (Horizontal Representation; Exhaustive Search). See Table 24 for details.

Supplementary Table 1 – continued from previous page

Acronym	**Full Name & Reference**	**Taxonomic Classification**
DIRAC	Differential Rank Conservation Eddy et al. (2010)	A system-level method assessing the stability of gene ranking within predefined pathways across phenotypes. (Meta-relational Representation; Exhaustive Analysis).
DTSP-V	Dynamic Top Scoring Pairs on Variance Wu et al. (2016)	A specialized TSP variant for time-series data using trend analysis as an alternative relation. (Horizontal Representation; Exhaustive Search). See Table 44 for details.
EMTTree+RX	Evolutionary Multi-Test Tree with RXA Czajkowski et al. (2025b)	An evolutionary-designed decision tree with multi-test nodes integrating univariate and pairwise rules. (Hierarchical Representation; Evolutionary Computation). See Table 40 for details.
EvoHDTree	Evolutionary Heterogeneous Decision Tree Czajkowski et al. (2021)	An evolutionary-designed decision tree with heterogeneous nodes allowing different test types. (Hierarchical Representation; Evolutionary Computation). See Table 38 for details.
Evo-REDT	Evolutionary Relative Expression Decision Tree Czajkowski et al. (2020a)	An evolutionary-designed decision tree using weighted gene-pair relations as splitting tests. (Hierarchical Representation; Evolutionary Computation). See Table 35 for details.
EvoTSP	Evolutionary Top-Scoring Pair Czajkowski and Kretowski (2014)	An evolutionary algorithm that optimizes a set of weighted gene pairs. (Horizontal Representation; Evolutionary Computation). See Table 32 for details.
GERs	Gene Expression Ratios Gordon et al. (2002)	A foundational method using simple gene expression ratios as an alternative relation. (Horizontal Representation; Exhaustive Search).
GTSPDT	Global TSP Decision Tree Czajkowski and Kretowski (2013)	An evolutionary-designed decision tree that globally optimizes the structure and selection of single-pair tests. (Hierarchical Representation; Evolutionary Computation). See Table 31 for details.
HC	Hierarchical Classification Tan et al. (2005)	A multi-class scheme using binary TSP or k-TSP classifiers at each node of a binary tree. (Hierarchical Representation; Exhaustive Search).

Supplementary Table 1 – continued from previous page

Acronym	**Full Name & Reference**	**Taxonomic Classification**
iPAGE	individualized Pairwise Analysis of Gene Expression Zheng et al. (2021)	A reference-based REO method for single-sample analysis by identifying reversed gene pairs against a reference. (Horizontal Representation; Exhaustive Search).
Ik-TSP	Improved k-Top Scoring Pair Zhou (2012)	An enhancement of k-TSP that uses an entropy-based method for feature pre-selection. (Horizontal Representation; Exhaustive Search). See Table 23 for details.
KF-k-TSP	Kernel Function k-Top Scoring Pair Li et al. (2023)	A k-TSP variant using non-linear kernel functions as an alternative relation. (Horizontal Representation; Exhaustive Search). See Table 29 for details.
k-GeneTriple	k-Gene Triple Yoon et al. (2008)	An extension of k-TSP using a voting ensemble of the top k gene triplets. (Horizontal Representation; Exhaustive Search). See Table 20 for details.
KNN+RRM	KNN+Relative Relation Metric Kartowicz-Stolarska and Czajkowski (2024)	A hybrid model embedding RXA logic into a custom distance metric for k-NN classifiers. (Embedded Representation; Exhaustive Search). See Table 45 for details.
k-TSN	k-Top Scoring N-tuples Yoon et al. (2010)	A generalization of k-TSP using a voting ensemble of the top k N-tuples. (Horizontal Representation; Heuristic Search). See Table 21 for details.
k-TSP	k-Top Scoring Pair Tan et al. (2005)	An extension of TSP that uses a voting ensemble of the top k disjoint gene pairs. (Horizontal Representation; Exhaustive Search). See Table 15 for details.
LC-k-TSP	Linear Combination k-Top Scoring Pair Lin et al. (2019)	A k-TSP variant where an SVM learns a unique linear decision boundary for each pair. (Horizontal Representation; Exhaustive Search). See Table 28 for details.
meGPS	multi-omics enhanced Gene Pair Signature Wu et al. (2022)	A multi-omics signature model applying gene-pair logic across different data types. (Horizontal Representation; Exhaustive Search). See Table 46 for details.

Supplementary Table 1 – continued from previous page

Acronym	**Full Name & Reference**	**Taxonomic Classification**
MIC	Maximal Information Coefficient Method Chen et al. (2016)	A method using the Maximal Information Coefficient as an alternative relation to discover synergistic gene pairs. (Horizontal Representation; Exhaustive Search). See Table 25 for details.
PenDA	Personalized Differential Analysis Richard et al. (2020)	A reference-based REO method for N-of-1 inference based on local gene-expression ordering against a reference. (Horizontal Representation; Exhaustive Search).
PIANOS	Platform Independent and Normalization-free One-Sample classifier Cai et al. (2025)	A supervised prognostic ensemble of k-TSP models trained on pathway-level scores. It is meta-relational in representation because relational rules are applied to pathway features rather than directly to individual genes. (Meta-relational/pathway-level Representation; Ensemble/Exhaustive Search). See Table 47 for details.
RankComp	Rank-based Comparative analysis Wang et al. (2015)	A reference-based REO method for detecting differentially expressed genes by analyzing order reversals. (Horizontal Representation; Exhaustive Search).
RankCompV3	RankComp version 3 Yan et al. (2024)	A reference-based REO extension for single-cell differential expression analysis using stable gene-pair orderings. (Horizontal Representation; Exhaustive Search).
REHA	Relative Evolutionary Hierarchical Analysis Czajkowski and Kretowski (2019)	An evolutionary-designed decision tree where nodes represent hierarchical gene clusters. (Hierarchical Representation; Evolutionary Computation). See Table 34 for details.
RMCT	Relative Multi-test Classification Tree Czajkowski et al. (2025a)	A top-down induced decision tree with "multi-test" nodes containing multiple TSP rules. (Hierarchical Representation; Greedy search; GPU Implementation). See Table 39 for details.
RRF	Random Rank Forest Lu et al. (2024)	An ensemble of rank-based decision trees, applying the Random Forest concept to relational splits. (Hierarchical Representation; Ensemble/Randomized Computation; Parallel (CPU) Execution). See Table 41 for details.

Supplementary Table 1 – continued from previous page

Acronym	Full Name & Reference	Taxonomic Classification
RXA-GSP	Relative Expression Analysis with Gene-set Pairs Yang et al. (2013)	A method that evaluates pairs of gene sets (pathways) for class discrimination, combining RXA logic with pathway knowledge. (Meta-relational Representation; Exhaustive Search).
RXCT	Relative Expression Classification Tree Czajkowski et al. (2020b)	A top-down induced decision tree using single gene-pair or triplet comparisons as splitting tests. (Hierarchical Representation; Greedy search; GPU Implementation). See Table 36 for details.
scPAGE	single-cell Pathway Analysis of Gene Expression Wang et al. (2022)	An adaptation of REO concepts to single-cell data, scoring pathway activity via order inversions. (Meta-relational Representation; Exhaustive Search).
SVM+k-TSP	Support Vector Machine with k-Top Scoring Pairs Shi et al. (2011)	A hybrid model using binary relational features from k-TSP as inputs to an SVM. (Embedded Representation; Exhaustive Search). See Table 22 for details.
TIGER	Top Inter-Gene Relations Czajkowski et al. (2016)	An evolutionary algorithm that searches for informative sets of weighted gene pairs and triplets. (Horizontal Representation; Evolutionary Computation). See Table 33 for details.
TSP	Top Scoring Pair Geman et al. (2004)	The foundational single-pair classifier. (Horizontal Representation; Exhaustive Search). See Table 14 for details.
TSPDT	Top-Scoring Pair Decision Tree Czajkowski and Kretowski (2011)	A top-down induced decision tree using a single top-scoring pair as the splitting test at each node. (Hierarchical Representation; Exhaustive Search). See Table 30 for details.
TSN	Top Scoring N-tuples Magis and Price (2012)	A generalization of TSP to a single N-gene combination (N-tuple). (Horizontal Representation; Exhaustive Search). See Table 18 for details.
TST	Top Scoring Triplet Lin et al. (2009)	An extension of TSP to a single triplet of genes. (Horizontal Representation; Exhaustive Search). See Table 17 for details.
VH-k-TSP	Vertical and Horizontal k-Top Scoring Pair Huang et al. (2018)	A k-TSP variant that integrates both within-sample and cross-sample comparisons. (Horizontal Representation; Exhaustive Search). See Table 26 for details.

Supplementary Table 1 – continued from previous page

Acronym	Full Name & Reference	Taxonomic Classification
WTSP	Weighted Top Scoring Pair Luo et al. (2008)	A TSP variant that extends the scoring mechanism to handle class imbalance. (Horizontal Representation; Exhaustive Search). See Table 16 for details.
WT-k-TSP	Weighted k-Top Scoring Pair Czajkowski and Kretowski (2008)	An extension of k-TSP that assigns weights to pairs based on expression ratios. (Horizontal Representation; Exhaustive Search). See Table 19 for details.

S2 - Review Methodology

This review was conducted as a structured narrative review with PRISMA-inspired transparency reporting rather than as a formal systematic review or meta-analysis. The objective was to capture methodological developments in rank-based relational analysis across supervised RXA, reference-based REO, and system-level rank-conservation approaches, and to organize them within the taxonomy proposed in the main manuscript.

S2.1 Search sources and search period

The review covered publications from January 2000 to May 2026, with the final formal search update performed on 31 May 2026. Database searches were performed in PubMed and Scopus. Google Scholar was used as a supplementary manual check for recent records, review articles, software resources, and forward citations of foundational studies, rather than as a fully reproducible database source. The search period was selected to cover early gene-pair and ratio-based classifiers, the formalization of Top-Scoring Pair methods, subsequent RXA extensions, REO-based single-sample methods, and system-level rank-conservation approaches. The final PubMed and Scopus searches were exported and archived using query identifiers. PubMed records were exported in `.nbib` format and Scopus records in bibliographic export format. The query identifiers used below correspond to the exported files retained by the authors.

S2.2 Search strings

PubMed searches were performed using Title/Abstract fields and explicit publication-date limits. Scopus searches used equivalent `TITLE-ABS-KEY` queries with year-based publication filters; the exports were generated during the final 31 May 2026 update. Google Scholar was used manually because it does not provide a stable and fully reproducible exportable database search.

PubMed queries. PM_Q1_TSP

```
(("top scoring pair"[Title/Abstract] OR "top-scoring pair"[Title/Abstract] OR "k-TSP"[Title/Abstract] OR "
    k top scoring pair"[Title/Abstract]) AND (gene OR transcriptomic OR transcriptomics OR expression OR
    omics OR biomarker OR diagnosis OR prognosis)) AND ("2000/01/01"[Date - Publication] : "2026/05/31"[
    Date - Publication])
```

PM_Q1_TSP_RECENT

```
(("top scoring pair"[Title/Abstract] OR "top-scoring pair"[Title/Abstract] OR "k-TSP"[Title/Abstract] OR "
    k top scoring pair"[Title/Abstract]) AND (gene OR transcriptomic OR transcriptomics OR expression OR
    omics OR biomarker OR diagnosis OR prognosis)) AND ("2025/01/01"[Date - Publication] : "2026/05/31"[
    Date - Publication])
```

PM_Q2_REO

```
(("relative expression ordering"[Title/Abstract] OR REO[Title/Abstract] OR RankComp[Title/Abstract] OR
    PenDA[Title/Abstract] OR iPAGE[Title/Abstract]) AND (gene OR transcriptomic OR omics OR "single
    sample" OR "single-sample")) AND ("2000/01/01"[Date - Publication] :
"2026/05/31"[Date - Publication])
```

PM_Q2_REO_RECENT

```
(("relative expression ordering"[Title/Abstract] OR REO[Title/Abstract] OR RankComp[Title/Abstract] OR
    PenDA[Title/Abstract] OR iPAGE[Title/Abstract]) AND (gene OR transcriptomic OR omics OR "single
    sample" OR "single-sample")) AND ("2025/01/01"[Date - Publication] : "2026/05/31"[Date - Publication
    ])
```

PM_Q3_DIRAC

```
(("differential rank conservation"[Title/Abstract] OR DIRAC[Title/Abstract] OR "rank conservation"[Title/
    Abstract] OR "pathway rank"[Title/Abstract]) AND (gene OR transcriptomic OR omics OR pathway OR
    module)) AND ("2000/01/01"[Date - Publication] : "2026/05/31"[Date - Publication])
```

PM_Q3_DIRAC_RECENT

```
(("differential rank conservation"[Title/Abstract] OR DIRAC[Title/Abstract] OR "rank conservation"[Title/
    Abstract] OR "pathway rank"[Title/Abstract]) AND (gene OR transcriptomic OR omics OR pathway OR
    module)) AND ("2025/01/01"[Date - Publication] : "2026/05/31"[Date - Publication])
```

PM_Q4_GENE_PAIR

```
(("gene pair"[Title/Abstract] OR "gene-pair"[Title/Abstract]) AND (classifier OR classification OR
    biomarker OR diagnosis OR
prognosis OR survival) AND (rank OR ordering OR "relative expression")) AND ("2000/01/01"[Date -
    Publication] : "2026/05/31"[Date - Publication])
```

PM_Q4_GENE_PAIR_RECENT

```
(("gene pair"[Title/Abstract] OR "gene-pair"[Title/Abstract]) AND (classifier OR classification OR
    biomarker OR diagnosis OR prognosis OR survival) AND (rank OR ordering OR "relative expression")) AND
     ("2025/01/01"[Date - Publication] : "2026/05/31"[Date - Publication])
```

PM_Q5_WITHIN_SAMPLE

```
(("within-sample ranking"[Title/Abstract] OR "within-sample ordering"[Title/Abstract] OR "within sample
    ranking"[Title/Abstract] OR "within sample ordering"[Title/Abstract]) AND (gene OR transcriptomic OR
    omics OR biomarker)) AND ("2000/01/01"[Date - Publication] : "2026/05/31"[Date - Publication])
```

PM_Q5_WITHIN_SAMPLE_RECENT

```
(("within-sample ranking"[Title/Abstract] OR "within-sample ordering"[Title/Abstract] OR "within sample
    ranking"[Title/Abstract] OR "within sample ordering"[Title/Abstract]) AND (gene OR transcriptomic OR
    omics OR biomarker)) AND ("2025/01/01"[Date - Publication]
: "2026/05/31"[Date - Publication])
```

Scopus queries. SC_Q1_TSP

```
TITLE-ABS-KEY("top scoring pair" OR "top-scoring pair" OR "k-TSP" OR "k top scoring pair") AND TITLE-ABS-
    KEY(gene OR transcriptomic OR transcriptomics OR expression OR omics OR biomarker OR diagnosis OR
    prognosis) AND PUBYEAR > 1999 AND PUBYEAR < 2027
```

SC_Q1_TSP_RECENT

```
TITLE-ABS-KEY("top scoring pair" OR "top-scoring pair" OR "k-TSP" OR "k top scoring pair") AND TITLE-ABS-
    KEY(gene OR transcriptomic OR transcriptomics OR expression OR omics OR biomarker OR diagnosis OR
    prognosis) AND PUBYEAR > 2024 AND PUBYEAR < 2027
```

SC_Q2_REO

```
TITLE-ABS-KEY("relative expression ordering" OR REO OR RankComp OR PenDA OR iPAGE) AND TITLE-ABS-KEY(gene
    OR transcriptomic OR omics OR "single sample" OR "single-sample") AND PUBYEAR
> 1999 AND PUBYEAR < 2027
```

SC_Q2_REO_RECENT

```
TITLE-ABS-KEY("relative expression ordering" OR REO OR RankComp OR PenDA OR iPAGE) AND TITLE-ABS-KEY(gene
    OR transcriptomic OR omics OR "single sample" OR "single-sample") AND PUBYEAR > 2024 AND PUBYEAR <
    2027
```

SC_Q3_DIRAC

```
TITLE-ABS-KEY("differential rank conservation" OR DIRAC OR "rank conservation" OR "pathway rank") AND
    TITLE-ABS-KEY(gene OR transcriptomic OR omics OR pathway OR module) AND PUBYEAR > 1999 AND PUBYEAR <
    2027
```

SC_Q3_DIRAC_RECENT

```
TITLE-ABS-KEY("differential rank conservation" OR DIRAC OR "rank conservation" OR "pathway rank") AND
    TITLE-ABS-KEY(gene OR transcriptomic OR omics OR pathway OR module) AND PUBYEAR > 2024 AND PUBYEAR <
    2027
```

SC_Q4_GENE_PAIR

```
TITLE-ABS-KEY(("gene pair" OR "gene-pair") AND (classifier OR classification OR biomarker
OR diagnosis OR prognosis OR survival) AND (rank OR ordering OR "relative expression")) AND PUBYEAR > 1999
     AND PUBYEAR < 2027
```

SC_Q4_GENE_PAIR_RECENT

```
TITLE-ABS-KEY(("gene pair" OR "gene-pair") AND (classifier OR classification OR biomarker OR diagnosis OR
    prognosis OR survival) AND (rank OR ordering OR "relative expression")) AND PUBYEAR > 2024 AND
    PUBYEAR < 2027
```

Google Scholar manual checks. Google Scholar was used only as a supplementary manual source because its ranking, coverage, and export options are not fully reproducible. The manual update focused on recent records and citation snowballing from foundational studies, recent reviews, and software resources; candidate records were included only when they satisfied the inclusion criteria below. Targeted queries included:

- `"Top Scoring Pair" gene expression classifier 2025 2026`
- `"k-TSP" biomarker 2025 2026`
- `"Relative Expression Ordering" RankComp 2025 2026`
- `"gene pair classifier" clinical research 2025`
- `"Differential Rank Conservation" omics 2025 2026`
- `"within-sample ordering" transcriptomics biomarker`
- `"rank-based" "gene-pair" "classifier" omics`
- `"single-sample" "relative expression ordering"`
- `"top scoring pair" "multi-omics"`
- `"rank-based" "spatial transcriptomics" "pathway"`

S2.3 Inclusion criteria

Publications were considered eligible when they met at least one of the following criteria:

- introduced a rank-based, ordering-based, pairwise, higher-order, pathway-level, or system-level relational method;
- extended an existing RXA, REO, or rank-conservation method computationally, statistically, or algorithmically;

- provided a relevant software implementation or benchmark of relational methods;
- represented a recent review or synthesis paper that helped position gene-pair, rank-based, REO, or pathway-level relational methods within the broader biomedical AI literature;
- applied relational methods to biomedical, clinical, omics, or translational data in a way that informed method selection, validation, or interpretation;
- provided methodological context for rank-based robustness, pairwise comparisons, pathway-level rank conservation, or single-sample inference.

S2.4 Exclusion criteria

We excluded publications when they met one or more of the following criteria:

- used the term "relative expression" only to describe conventional differential expression or fold-change analysis without a rank-based or relation-based model;
- focused on general machine-learning or statistical methods without a clear connection to within-sample ordering, pairwise feature relations, rank conservation, or relational modelling;
- used terms such as "rank", "ranking", or "DIRAC" in unrelated contexts, including generic feature ranking, search-result ranking, physics/materials-science uses of the Dirac acronym, or other non-biomedical meanings;
- lacked sufficient methodological detail to classify the approach within the proposed taxonomy;
- reported only a biological application without methodological relevance to relational analysis;
- duplicated another record, conference abstract, or preliminary version without adding relevant methodological information.

S2.5 Screening and synthesis procedure

The search was documented at the query level. For each PubMed and Scopus query, the number of raw query hits was recorded and the corresponding bibliographic export was retained. Because the same record could be retrieved by multiple queries and by both databases, counts are reported as raw query hits before deduplication unless stated otherwise. Screening proceeded in three stages. First, obvious duplicates and clearly unrelated records were removed, including many false-positive Scopus records retrieved by the broad DIRAC/rank-conservation query. Second, titles and abstracts were screened for relevance to within-sample ordering, pairwise or higher-order feature relations, REO, DIRAC/rank conservation, pathway-level relational reasoning, software availability, or clinically informative applications. Third, records that appeared relevant or ambiguous were assessed in more detail using the full text or detailed bibliographic record where available. For each included or borderline method, we extracted the method name, acronym, primary reference, data modality, application domain, relational representation, computational strategy, implementation/software status, validation context, and taxonomic placement where available. Application-only papers were not used to define new method families unless they introduced a substantial methodological or computational extension. The extracted information was used to construct the taxonomic glossary, complexity summary, software overview, qualitative comparison, recent-update table, and benchmark discussion. The review intentionally combines database search with citation snowballing because

relevant methods are known under heterogeneous terminology, including "gene-pair signatures", "relative expression", "rank conservation", "single-sample analysis", "within-sample ordering", and "pairwise molecule comparisons". This broader strategy was necessary to reduce the risk of missing methodological families that developed in parallel but used different terminology.

S2.6 PRISMA-inspired flow summary

The search counts are summarized in Supplementary Table 2, and the screening process is summarized in Supplementary Table 3. Because the study is a structured narrative review rather than a formal systematic review or meta-analysis, the flow table should be interpreted as a transparency aid rather than as a full PRISMA claim. Counts are exact for raw PubMed and Scopus query hits and approximate after the raw-search stage because database overlap, Google Scholar manual checking, and citation snowballing do not produce a single reproducible deduplicated record set.

Supplementary Table 2: Raw PubMed and Scopus query hits recorded during the final literature update. Counts are raw query hits before deduplication; the same publication may appear in multiple queries or databases. Recent counts refer to the targeted 2025–2026 update.

Query family	**PubMed full**	**PubMed recent**	**Scopus full**	**Scopus recent**
TSP / k-TSP	62	3	113	10
REO / RankComp / PenDA / iPAGE	120	14	202	19
DIRAC / rank conservation / pathway rank	141	34	627	120
Gene-pair classifier / signature	86	6	148	14
Within-sample ranking / ordering	0	0	–	–
Total raw query hits	**409**	**57**	**1090**	**163**

Combined PubMed + Scopus totals: 1499 full-period raw query hits and 220 recent-update raw query hits.

The DIRAC/rank-conservation Scopus query intentionally used broad terms to avoid missing biomedical rank-conservation studies, but it retrieved many unrelated records in which "DIRAC" referred to physics, materials science, or unrelated computational concepts. These records were excluded during title/abstract screening.

Supplementary Table 3: PRISMA-inspired summary of literature identification and screening. Counts after the raw database-search stage are approximate because the review combined bibliographic database searches, Google Scholar manual checking, and citation snowballing.

Screening stage	**Number of records**
Raw PubMed and Scopus query hits before deduplication and relevance filtering	1499
Raw query hits retrieved by the targeted 2025–2026 update	220
Records retained after duplicate removal, removal of obvious false-positive query hits, and relevance prioritization	> 150
Records screened in detail by title/abstract or detailed bibliographic record	> 150
Records excluded as clearly unrelated, duplicate, outside the biomedical/omics/AI scope, or outside the relational-methods scope	approximately 50–60
Full-text articles or detailed records assessed for methodological, software, benchmark, or translational relevance	approximately 100–110
Records excluded after full-text/detailed assessment because they lacked a rank-/relation-based method, provided insufficient methodological detail, duplicated another source, or represented application-only use without methodological relevance	approximately 10–20
Core publications included in the review synthesis	approximately 100

S2.7 Quality and relevance assessment

We did not perform a formal risk-of-bias assessment because the review covers heterogeneous methodological, software, benchmark, review, and application studies rather than a homogeneous set of clinical trials, diagnostic accuracy studies, or intervention studies. Instead, each publication was assessed for relevance to the taxonomic and methodological aims of the review. The following criteria were considered:

- **Methodological relevance:** whether the publication introduced or substantially extended a rank-based, ordering-based, pairwise, higher-order, pathway-level, or system-level relational method.
- **Clarity of relational representation:** whether the relation object was explicit, e.g. a pairwise ordering $x_i > x_j$, a weighted or parameterized relation, a pathway-level rank-conservation score, or an embedded vector of relational features.
- **Computational reproducibility:** whether the algorithmic procedure, scoring rule, search strategy, software implementation, or benchmark protocol was sufficiently described to classify the method.
- **Software and resource availability:** whether code, package, web resource, or executable implementation was available or described.
- **Validation design:** whether the study used cross-validation, independent cohorts, external validation, cross-platform testing, reference-cohort validation, or only retrospective/internal evaluation.
- **Taxonomic relevance:** whether the record helped distinguish supervised RXA, reference-based REO, system-level rank conservation, embedded/hybrid relational methods, or borderline pairwise interpretable ML approaches.
- **Clinical or translational relevance:** whether application papers illustrated realistic use cases, limitations, or validation requirements without necessarily introducing a new method family.

Recent review papers and application studies were included only when they informed the taxonomy, documented methodological trends, provided representative clinical examples, or highlighted practical limitations. Application-only papers were generally treated as supplementary examples unless they introduced a new method family or a substantial computational extension. When evidence was limited to retrospective datasets, narrow application domains, unavailable software, or non-independent validation, this was treated as a limitation in the qualitative comparison and discussion.

S2.8 Recent-update screening decisions

To document the final 2025–2026 literature update, Supplementary Table 4 summarizes representative recent records and how they were used in the review. This table is not an exhaustive list of all screened records. It documents records that were considered relevant enough to affect the manuscript, Online Resource, bibliography, or exclusion rationale.

Supplementary Table 4: Representative recent records considered during the final literature update. The table documents records that affected the manuscript, Online Resource, bibliography, or exclusion rationale; it is not an exhaustive list of all screened 2025–2026 records.

Record	**Use in review**	**Reason for inclusion or treatment**
Wu et al., gene-pair methods in clinical research Wu et al. (2025)	Main text / background	Recent review directly relevant to gene-pair methods in clinical and transcriptomic settings. Used to position the review relative to existing gene-pair-focused surveys.
Zheng et al., relative rank for compositional NGS data Zheng et al. (2025)	Main text / background	Supports the broader motivation for rank-based analysis in compositional and sequencing-derived omics data, but is not treated as a core RXA method.
Cai et al., PIANOS Cai et al. (2025)	Main text / supplement	Borderline but important case linking supervised k-TSP logic with pathway-level prognostic modelling. Treated as a supervised pathway-level relational ensemble, not as task-equivalent REO or DIRAC.
Tian et al., pre-analytical variables and REOs Tian et al. (2025)	Limitations / supplement	Relevant to robustness claims. Used to support cautious wording about pre-analytical variation, platform effects, and the conditional nature of rank-based robustness.
Zheng et al., TB-Scope Zheng et al. (2025)	Application example	Multicenter TSP-ensemble application for active tuberculosis diagnosis across microarray and RNA-seq data. Included as an important application example, not as a new taxonomic branch.
Fourquet et al., bivariate monotonic classifiers Fourquet et al. (2025)	Supplement / AI-ML context	Relevant to compact pairwise interpretable modelling, but not a canonical RXA, REO, or DIRAC-style method. Used as broader AI/ML context.
Huang et al., multiple forms of pairwise molecule comparisons Huang et al. (2026)	Supplement / uncertain	Potentially relevant extension of pairwise-comparison logic. Treated cautiously because full methodological placement requires additional verification before considering it a core method.
Ambrosini et al., 8-gene k-TSP-related signature in colorectal cancer immunotherapy Ambrosini et al. (2026)	Supplement only	Recent clinical application of k-TSP-related gene-pair logic. Included as evidence of continued applied use, but it does not introduce a new method family.
Tercan, robust predictors for AML drug response Tercan (2026)	Supplement only	Recent application of k-TSP-style predictors in drug-response analysis. Useful as an applied example, but not a new taxonomic branch.
Qian et al., pathological response prediction in rectal cancer Qian et al. (2025)	Supplement only	Relevant application of REO-like or gene-pair logic with ensemble learning. Included as a recent applied example, not as a new method family.

S2.9 Limitations of the review methodology

The review is not a meta-analysis and does not estimate pooled effect sizes. It should be interpreted as a structured narrative review with PRISMA-inspired transparency reporting, not as an exhaustive systematic review. PubMed and Scopus searches were reproducible at the query level. Google Scholar was used only as a supplementary manual check because its ranking, coverage, and export behaviour are not stable. Relevant papers indexed outside the searched databases may have been missed. The terminology in this field is heterogeneous, and some methods may not explicitly use RXA, REO, TSP, DIRAC, or rank-conservation terminology despite relying on related relational principles. Conversely, broad terms such as "rank", "ranking", "relative expression", and "DIRAC" retrieved many false-positive records, including conventional differential-expression studies, generic feature-ranking papers, and non-biomedical uses of the DIRAC acronym. To mitigate these limitations, we combined database searches with citation snowballing, cross-checked recent surveys and software resources, and manually reviewed recent 2025–2026 records for methodological or representative translational relevance.

S3 - Computational Complexity of Representative Algorithms

This section summarizes the theoretical time complexity of selected rank-based relational algorithms. These complexities illustrate the main computational trade-offs inherent in different approaches. The primary challenge stems from the combinatorial explosion of potential feature relationships: a dataset with P features contains $O(P^2)$ possible pairs, $O(P^3)$ triplets, and so on. The complexity figures presented in Supplementary Table 5 generally assume naive, exhaustive search implementations. In practice, many algorithms can be significantly optimized. A common strategy to mitigate the computational load is to incorporate pre-filtering steps, such as variance thresholds or differential expression screening, which reduce the number of candidate features P before the main relational search begins. While such steps improve efficiency, they risk discarding informative relations relevant to subtle or combinatorial effects. In predictive benchmarks, any supervised or data-dependent filtering must be performed within the training split or inner validation loop to avoid information leakage.

Supplementary Table 5: Approximate dominant training-time complexity of representative rank-based relational algorithms.

Algorithm	**Time Complexity**
Top-Scoring Pair (TSP) Geman et al. (2004)	$O(P^2 \cdot M)$
k-Top-Scoring Pairs (k-TSP) Tan et al. (2005)	$O(P^2 \cdot M)$
Top-Scoring Triplet (TST) Lin et al. (2009)	$O(P^3 \cdot M)$
Top-Scoring N-tuple (TSN) Magis and Price (2012)	$O(P^g \cdot M)$
Relative Expression Ordering (RankComp) Wang et al. (2015)	$O(P^2 \cdot M)$
TSP Decision Tree (TSPDT) Czajkowski and Kretowski (2011)	$O(T \cdot P^2 \cdot M)$
Evolutionary RXA (EvoTSP) Czajkowski and Kretowski (2014)	$O(G \cdot \lambda \cdot C)$
Evolutionary RXA Tree (Evo-REDT) Czajkowski et al. (2020a)	$\sim O(G \cdot \lambda \cdot \text{nodes})$

The variables in the complexity notations are defined as follows: P is the number of features, M is the number of samples, g is the tuple size in TSN-like methods, and T is the number of non-terminal nodes in a decision tree such as TSPDT. For evolutionary algorithms, G denotes the number of generations, λ the population size, and C the cost of a single fitness evaluation, which may itself include cross-validation, relation scoring, or tree evaluation. The complexity for Evo-REDT is given as an approximation, reflecting its non-exhaustive heuristic search

and dependence on the evolving tree structure. The complexity values in Supplementary Table 5 describe dominant training-time behaviour and should be interpreted as order-of-growth summaries rather than exact runtime predictions. Actual runtime depends on implementation language, memory layout, parallelization strategy, feature filtering, candidate caching, and whether all candidate scores are materialized or streamed. In many practical implementations, memory rather than arithmetic operations becomes the limiting factor when P is large, because the number of pairwise scores grows quadratically and higher-order candidate sets grow combinatorially.

S4 - Evolutionary Search Strategies

Evolutionary Algorithms (EAs) are metaheuristic optimization methods inspired by natural selection, designed to search complex solution spaces by evolving a population of candidate solutions through selection, mutation, and crossover Liu et al. (2024); Michalewicz (1996). In the context of RXA, EAs have been used to explore large spaces of possible feature relations without exhaustively evaluating each candidate. They provide a stochastic optimization strategy that may reduce dependence on greedy local choices, but they do not guarantee global optimality and require careful validation, parameter control, and repeated-run assessment. This section summarizes the main evolutionary strategies employed in RXA-based models to optimize feature relations, classifier structures, or hybrid decision architectures.

1. **Direct Optimization of Feature Relations:** This category focuses on directly optimizing sets of feature relations used for classification. These methods evolve populations of candidate gene pairs or more complex constructs, where fitness is typically a function of classification accuracy and model complexity.

    - **EvoTSP** Czajkowski and Kretowski (2014): Extends the k-TSP method by dynamically selecting and weighting gene pairs. Each individual in the population represents a group of gene pairs with associated weights, which are optimized through evolutionary processes. The fitness function balances classification accuracy with relation diversity.
    - **TIGER** Czajkowski et al. (2016): Expands on the concept of EvoTSP by evolving populations of variable-sized gene groups, including pairs and triplets. It uses specialized crossover and mutation operators to search for higher-order relational patterns.

2. **Evolution of Decision Structures:** This category leverages EAs to evolve entire tree-like structures where internal nodes are based on rank-based gene relations. This combines the hierarchical representation with the power of global evolutionary optimization.

    - **GTSPDT** Czajkowski and Kretowski (2013): Generalizes TSP-based decision trees by using an evolutionary algorithm to globally optimize both the tree topology and the selection of gene pairs at each node, avoiding the greedy, locally optimal choices of top-down induction.
    - **REHA** Czajkowski and Kretowski (2019): Introduces hierarchical gene groupings at internal nodes. The evolutionary process optimizes both the cluster composition and the decision rules, aiming to find coherent groups of co-expressed genes.
    - **Evo-REDT** Czajkowski et al. (2020a): Induces decision trees where splitting nodes are based on weighted gene pairs. The evolutionary algorithm optimizes both the tree structure and the weights of the relations to maximize classification quality.
    - **EMTTree+RX** Czajkowski et al. (2025b): Integrates both univariate and TSP-based tests into "multi-test" nodes within an RXA tree. The EA optimizes the

combination of tests at each node and the overall tree architecture, aiming to improve performance while retaining an auditable tree structure.

3. **Hybrid and Parallel Evolutionary Frameworks:** To manage the computational cost of EAs, advanced frameworks have been developed.
   - Some evolutionary RXA models use hybrid CPU–GPU parallelization. In this approach, computationally intensive tasks such as relation scoring or fitness evaluation can be offloaded to GPU threads, while CPUs manage higher-level population dynamics.
   - Fitness functions in these models may combine classification performance with other criteria such as model sparsity, relation diversity, tree size, or the number of distinct genes used in the model.

Stochastic search introduces an additional reproducibility dimension. Reported performance is most credible when model selection is separated from final evaluation, for example through nested validation or independent test cohorts, and when repeated runs with different random seeds are summarized rather than relying on a single evolutionary run. Hyperparameter settings, stopping criteria, population size, mutation and crossover operators, and random seeds should therefore be documented when evolutionary relational models are benchmarked or compared.

S5 - Practical Guidance for Applying RXA Methods

S5.1 - Practical Considerations and Recommendations

This section expands on the guidance provided in the main manuscript, offering more detailed technical advice for practitioners applying RXA methods.

Data Preprocessing and Batch Effects. RXA methods can reduce sensitivity to some sample-level monotonic transformations and normalization choices, but they are not inherently immune to technical variation. Severe batch effects, feature-specific biases, and platform-dependent detection limits can introduce systematic, non-biological variation that alters gene rankings Wang et al. (2023); Yu et al. (2023). Before applying RXA, it is advisable to inspect the data for strong batch effects using methods such as Principal Component Analysis (PCA). If significant batch effects are present, specialized correction tools such as BIRCH Sundararaman et al. (2023) or PRPS Molania et al. (2023) may be considered. Additionally, handling missing values is important; methods that can accommodate missing ranks or imputation strategies may be necessary. Ties, missingness, and zero inflation should be assessed before ranking. If many features are exactly zero or near the detection limit, strict pairwise ordering may be dominated by technical sparsity rather than biology. Possible mitigation strategies include removing features with excessive missingness or zero counts, applying modality-appropriate imputation, using pseudo-bulk aggregation in single-cell data, introducing a margin below which a relation is treated as inactive, or using uncertainty-aware relation scores. These decisions should be made inside the training data during predictive evaluation to avoid information leakage.

Feature Filtering. In datasets with tens of thousands of features, exhaustive pairwise search can still be computationally expensive Magis and Price (2012). A common practical strategy is to pre-filter features to a more manageable number (e.g., 1,000–5,000) before applying RXA Budhraja et al. (2023); Zheng et al. (2024). This can be done using unsupervised variance-based filtering, train-only univariate statistical filters, or prior biological knowledge. In predictive benchmarks, feature filtering must be performed within each training split or within the inner loop of a nested validation design. Global filtering before cross-validation can leak information from the test fold and inflate performance estimates.

Handling Class Imbalance and Multi-Class Problems. Many foundational RXA methods assume balanced, binary classification problems. In real-world scenarios with severe class imbalance, consider using methods specifically designed to handle this, such as Weighted TSP (WTSP) Luo et al. (2008), or employing resampling strategies. For multi-class problems, one must either use decomposition schemes (e.g., one-vs-rest) or adopt an algorithm natively designed for multi-class classification.

Computational Resources and Hyperparameter Tuning. The choice of method is often constrained by available computational infrastructure. GPU-accelerated and evolutionary algorithms have reported competitive performance in selected studies Czajkowski et al. (2021); Mirza et al. (2019), but they require specialized hardware or substantial computational time. If such resources are unavailable, simpler methods such as k-TSP may remain a practical baseline. Furthermore, complex methods often have numerous hyperparameters that require careful tuning, ideally within a nested validation design.

S5.2 - Available Tools and Software

Supplementary Table 6 provides a summary of identified software packages and tools for selected rank-based relational methods. Availability status may change over time; the table should be interpreted as a practical resource guide rather than a complete software registry.

Supplementary Table 6: Publicly available software resources for selected rank-based relational methods. Availability reflects the information identified during the review and may change over time.

Tool/Package	Algorithms	Technology	Availability
`tspair` Leek (2009)	TSP	R	CRAN
`switchBox` Afsari et al. (2015)	k-TSP, TSP	R	CRAN
`multiclassPairs` Marzouka et al. (2021)	Multi-class k-TSP	R	CRAN
`MetaKTSP` Kim et al. (2016)	Meta-analytic k-TSP	R	GitHub
`ranktreeEnsemble` Lu et al. (2024)	Rank-Based Classification Trees	R	CRAN
`BTSP` Martínez-Cantín et al. (2017)	Bayesian TSP	R	GitHub
`pyRRMs` Kartowicz-Stolarska and Czajkowski (2024)	Relative Relation Metric	Python	GitHub
ITree Sokołowski et al. (2024)	Interactive DT with RXA	Website	Online
AUREA Earls et al. (2013)	TSP, k-TSP, TST	R	GitHub
TSN Algorithm Magis and Price (2012)	Top-Scoring N-tuples	MATLAB	Request
k-TSP with SVM Shi et al. (2011)	k-TSP + SVM	MATLAB	Request
MIC Method Chen et al. (2016)	MIC-Based Gene Selection	C++/MATLAB	Request

S5.3 - Reproducibility-Oriented Benchmark of Executable Methods

This section documents the harmonized benchmark used to generate Fig. 4 in the main manuscript. The benchmark was designed as a reproducibility-oriented comparison of executable supervised relational/rank-based methods and standard ML baselines, not as a definitive ranking of all relational algorithms. It is restricted to tools that could be wrapped into a common train/test interface and run on the same eight public gene-expression datasets.

Repository and availability. The complete benchmark artifact is available at `https://github.com/mczajkowskipb/AIR-relational-benchmark`. The repository contains the final benchmark matrices, fold definitions, method wrappers, scripts for rerunning the main protocol, result summaries, figure-generation scripts, and reviewer-facing documentation. The main result files are `results/summary/harmonized_cv5x5_f200_complete_overall_summary.csv` and `results/summary/harmonized_cv5x5_f200_complete_dataset_method_summary.csv`.

Primary protocol. The final benchmark protocol is denoted `cv_5x5_f200`. It uses five repeats of five-fold stratified cross-validation on eight public datasets. For each train/test split, MAD feature filtering is fitted only on the training samples and selects up to 200 features; the selected feature set and any fold-specific preprocessing estimated from the training split are then applied to the corresponding test fold. The same fold definitions are used for all methods. Balanced accuracy is the primary metric; macro-F1 and MCC are reported as secondary metrics. The final benchmark contains 3400 completed method–dataset–fold/repeat jobs and no failed jobs in the final result set. Aggregate results are computed as unweighted means of dataset-level method summaries across the eight datasets, rather than as pooled sample-level estimates. The SD BAC column reports the standard deviation of dataset-level balanced accuracy values across datasets, expressed in percentage points.

Scope and limitations. The benchmark includes executable supervised relational/rank-based wrappers, embedded relational classifiers, a rank-based ensemble, a relative-relation-metric kNN model, and standard ML baselines. It does not include REO and system-level rank-conservation methods as task-equivalent classifiers because these methods address different inferential objectives. Some Python gene-pair wrappers are benchmark-compatible implementations inspired by external notebook logic rather than verbatim reproductions of notebook workflows. The TSP+SVM and REO+SVM wrappers are closely related technical encodings followed by the same linear SVM classifier. Their nearly identical aggregate results should not be interpreted as independent confirmation of two distinct methodological families; rather, they indicate that, under this supervised cohort-level protocol, similar pairwise/rank encodings can lead to very similar SVM decision endpoints. The RRF result corresponds to the executable `ranktreeEnsemble::rforest` wrapper under the common f200 protocol, i.e. train-fold-only MAD filtering to at most 200 features, and should not be interpreted as an exhaustive hyperparameter tuning study of the original RRF method. In this benchmark, this wrapper produced the weakest non-majority aggregate result, illustrating that accessibility and ensemble capacity do not by themselves guarantee strong predictive performance under a shared protocol.

Dataset summary. The eight datasets are the same benchmark datasets used in the main manuscript: GDS2771, GSE10072, GSE17920, GSE19804, GSE25837, GSE27272, GSE3365, and GSE6613. Final matrices, dataset metadata, and fixed folds are provided in the repository; dataset-level balanced-accuracy, macro-F1, and MCC results are reported in Supplementary Tables 9–11.

Supplementary Table 7: Completed method families included in the harmonized benchmark. The grouping follows model form and explanation endpoint; it is descriptive and is not intended as a calibrated interpretability score.

Group	Displayed method	Implementation / wrapper	Main figure
Direct/structured relational	TSP	`switchBox` TSP wrapper	yes
Direct/structured relational	k-TSP	`switchBox` k-TSP wrapper	yes
Direct/structured relational	REO-like	Python gene-pair wrapper inspired by external notebook logic	yes
Direct/structured relational	TSPDT	`BigTSP` TSP-tree wrapper; run at f200 with increased R protection stack	yes
Embedded/hybrid relational	TSP+SVM	TSP-like pairwise/rank encoding followed by a linear SVM	yes
Embedded/hybrid relational	REO+SVM	REO-like pairwise relation encoding followed by a linear SVM	yes
Embedded/hybrid relational	TSP+RF	TSP-like pair encoding followed by random forest	yes
Embedded/hybrid relational	RRM-kNN	`pyrrm` relative-relation metric inside kNN, k=5	yes
Rank-based ensemble	RRF	`ranktreeEnsemble::rforest` wrapper under common f200 protocol	yes
Standard ML baseline	Elastic-net LR	`glmnet` wrapper	yes
Standard ML baseline	Linear SVM	`e1071` linear SVM wrapper	yes
Standard ML baseline	Euclidean kNN	standard value-space kNN baseline	yes
Standard ML baseline	CART tree	`rpart` decision tree baseline	yes
Standard ML baseline	Random Forest	`ranger` random forest baseline	yes
Standard ML baseline	XGBoost	shallow/default `xgboost` baseline	yes
Technical reference	Python TSP-like	Alternative Python single-pair wrapper; retained in supplementary tables but omitted from main figure to avoid duplicating TSP	no
Technical reference	Majority	majority-class sanity baseline	no

Supplementary Table 8: Aggregate performance of all completed methods under the harmonized benchmark. Values are means across the eight datasets; balanced accuracy is the primary metric.

Method	BAC (%)	Macro-F1 (%)	MCC	SD BAC (pp)
Linear SVM	72.4	71.9	0.449	19.7
Random Forest	72.1	71.0	0.465	19.0
Elastic-net LR	70.8	67.2	0.423	22.2
XGBoost	70.7	69.4	0.429	19.7
TSP+RF	69.7	68.5	0.400	21.1
RRM-kNN	69.3	68.0	0.385	21.0
TSP+SVM	68.5	68.2	0.373	21.6
REO+SVM	68.5	68.2	0.373	21.6
TSPDT (BigTSP)	68.2	67.8	0.368	19.2
Euclidean kNN	68.0	66.8	0.361	20.4
CART tree	67.7	67.4	0.362	18.4
REO-like	67.6	65.3	0.359	22.3
k-TSP (switchBox)	67.4	65.8	0.354	21.8
Python TSP-like	66.9	64.5	0.340	21.2
TSP (switchBox)	66.9	64.4	0.339	21.2
RRF (ranktreeEnsemble)	65.7	62.4	0.332	16.9
Majority	50.0	37.9	0.000	0.0

Supplementary Table 9: Dataset-level balanced accuracy (%) for all benchmarked methods under the harmonized `cv_5x5_f200` protocol.

Method	**GDS2771**	**GSE10072**	**GSE17920**	**GSE19804**	**GSE25837**	**GSE27272**	**GSE3365**	**GSE6613**
TSP (switchBox)	59.1	95.7	58.0	95.0	42.6	50.1	83.3	51.3
k-TSP (switchBox)	65.5	97.3	54.5	94.3	38.5	47.5	82.6	58.9
REO-like	62.2	97.8	54.2	95.8	34.5	54.0	82.6	59.7
TSPDT (BigTSP)	62.8	95.7	55.0	93.5	51.3	47.7	81.2	58.5
TSP+SVM	64.7	97.8	58.0	95.0	39.2	47.3	83.7	62.1
REO+SVM	64.7	97.5	58.0	95.7	39.2	47.3	83.4	62.1
TSP+RF	68.5	97.8	56.2	95.8	43.9	49.9	86.6	59.1
RRM-kNN	66.6	98.0	52.5	94.8	47.9	47.0	86.5	61.3
RRF (ranktreeEnsemble)	59.4	87.5	57.5	92.8	48.8	50.8	73.8	54.8
Elastic-net LR	72.8	98.6	54.6	95.5	50.0	49.9	94.2	51.1
Linear SVM	70.0	98.4	62.2	92.2	48.0	57.5	94.3	56.9
Euclidean kNN	68.1	98.0	52.0	93.7	48.4	46.4	79.8	57.6
CART tree	61.3	93.1	63.4	91.3	49.1	47.6	81.2	54.4
Random Forest	69.3	98.2	66.7	95.2	48.9	52.2	86.3	59.9
XGBoost	69.9	95.7	64.7	95.2	47.6	50.6	86.9	54.9
Python TSP-like	59.1	96.4	58.0	95.0	42.6	50.4	82.6	51.3
Majority	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0

Supplementary Table 10: Dataset-level macro-F1 (%) for all benchmarked methods under the harmonized cv_5x5_f200 protocol.

Method	**GDS2771**	**GSE10072**	**GSE17920**	**GSE19804**	**GSE25837**	**GSE27272**	**GSE3365**	**GSE6613**
TSP (switchBox)	58.1	95.6	54.9	95.0	37.3	45.5	80.0	49.3
k-TSP (switchBox)	64.9	97.3	52.7	94.3	34.8	44.2	80.8	57.0
REO-like	61.5	97.9	51.3	95.8	32.2	46.3	80.2	57.3
TSPDT (BigTSP)	62.5	95.6	54.3	93.5	50.4	47.5	80.9	57.8
TSP+SVM	64.6	97.9	57.3	95.0	38.3	47.0	83.6	61.8
REO+SVM	64.6	97.5	57.3	95.7	38.3	47.0	83.3	61.8
TSP+RF	68.2	97.9	55.8	95.8	40.0	45.3	86.4	58.7
RRM-kNN	66.2	98.1	51.3	94.8	44.7	44.2	83.8	61.0
RRF (ranktreeEnsemble)	54.7	86.2	55.9	92.7	42.8	45.3	73.4	48.2
Elastic-net LR	72.6	98.7	50.9	95.5	43.0	41.1	93.7	42.2
Linear SVM	69.8	98.5	61.5	92.2	47.0	57.1	93.7	55.8
Euclidean kNN	67.7	98.1	49.9	93.6	46.3	43.5	78.2	57.0
CART tree	60.9	93.0	63.4	91.3	47.7	47.4	81.8	53.4
Random Forest	69.1	98.3	68.2	95.2	42.4	48.6	87.1	59.5
XGBoost	69.7	95.8	65.7	95.1	42.2	44.9	87.1	54.2
Python TSP-like	58.1	96.4	54.9	95.0	37.3	45.8	79.5	49.4
Majority	34.7	35.1	41.4	33.3	43.0	41.2	40.1	34.4

Supplementary Table 11: Dataset-level Matthews correlation coefficient (MCC) for all benchmarked methods under the harmonized `cv_5x5_f200` protocol.

Method	**GDS2771**	**GSE10072**	**GSE17920**	**GSE19804**	**GSE25837**	**GSE27272**	**GSE3365**	**GSE6613**
TSP (switchBox)	0.193	0.919	0.151	0.903	-0.129	0.004	0.636	0.032
k-TSP (switchBox)	0.325	0.950	0.090	0.890	-0.209	-0.047	0.638	0.198
REO-like	0.254	0.961	0.079	0.919	-0.275	0.090	0.630	0.213
TSPDT (BigTSP)	0.260	0.919	0.107	0.874	0.020	-0.043	0.633	0.172
TSP+SVM	0.298	0.961	0.163	0.904	-0.223	-0.052	0.683	0.249
REO+SVM	0.298	0.954	0.163	0.917	-0.223	-0.052	0.678	0.249
TSP+RF	0.389	0.961	0.153	0.920	-0.154	0.001	0.739	0.188
RRM-kNN	0.345	0.964	0.065	0.898	-0.049	-0.076	0.702	0.232
RRF (ranktreeEnsemble)	0.219	0.774	0.152	0.864	-0.029	0.018	0.552	0.106
Elastic-net LR	0.467	0.975	0.131	0.912	0.000	-0.004	0.880	0.021
Linear SVM	0.407	0.971	0.245	0.846	-0.045	0.148	0.880	0.145
Euclidean kNN	0.377	0.965	0.049	0.877	-0.025	-0.094	0.580	0.157
CART tree	0.232	0.867	0.286	0.835	-0.024	-0.046	0.656	0.090
Random Forest	0.398	0.968	0.444	0.906	-0.041	0.087	0.755	0.205
XGBoost	0.408	0.921	0.394	0.906	-0.070	0.015	0.758	0.097
Python TSP-like	0.193	0.933	0.151	0.903	-0.129	0.011	0.625	0.030
Majority	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

S6 - Qualitative Comparison and Rating Rubric

This section provides a qualitative comparison of the algorithms discussed in the main manuscript. Its purpose is to support method selection by summarizing practical strengths and limitations, not to provide a formal meta-analysis or universal ranking of methods. The evidence base is heterogeneous: some algorithms have public software and multiple benchmark studies, whereas others have been evaluated only in narrow experimental settings or lack accessible implementations. Therefore, the ratings should be interpreted as evidence-informed qualitative categories rather than reproducible numerical scores. Each method is evaluated across five criteria: accuracy potential (AC), speed/scalability (SP), interpretability (IN), availability (AV), and applicability (AP). The three-level rating system is defined in Supplementary Table 12. A high rating indicates stronger evidence or practical utility within the reviewed context; it does not imply superiority across all datasets, modalities, or clinical settings. When evidence was limited, unavailable, or strongly task-specific, this limitation was reflected in the rating and in the accompanying justification.

The summary comparison in Supplementary Table 13 highlights broad practical trade-offs across algorithms. The subsequent detailed tables provide method-specific justifications. To reduce overstatement, terms such as "higher accuracy", "broad applicability", or "strong interpretability" refer only to the reviewed evidence and benchmark settings, not to universal superiority. Because published evidence is heterogeneous and most methods were not evaluated under a shared independent protocol, the accuracy-potential column is intentionally conservative and coarse-grained. It distinguishes methods with limited reported predictive evidence from methods with favourable results in selected settings, but it should not be interpreted as a fine-grained ranking of predictive performance.

The ratings in Supplementary Table 13 should be read together with the detailed justifications in the following tables. Availability refers to the software resources identified during the review and may change over time. A low or medium rating should not be interpreted as evidence that a method is weak; it may instead reflect limited software availability, narrow validation evidence, task-specific applicability, or insufficient reproducibility information.

Supplementary Table 12: Rubric used for the qualitative comparison of relational algorithms. The ratings are intended to support practical orientation and should not be interpreted as formal meta-analytic estimates.

Criterion	**Low (*)**	**Medium (**)**	**High (***)**
Accuracy potential (AC)	Limited or inconsistent reported performance; evidence restricted to very narrow settings; no clear advantage over simpler alternatives.	Competitive or moderately improved reported performance in selected datasets or tasks, but with limited external evidence or mixed results.	Favourable reported performance across multiple relevant benchmark or application settings, while recognizing that performance remains task- and dataset-dependent.
Speed / scalability (SP)	Computationally demanding for typical high-dimensional omics data; requires heavy tuning, specialized hardware, or impractical exhaustive search.	Feasible with filtering, moderate optimization, or parallelization; computational cost may still be relevant for large P or repeated validation.	Computationally lightweight or scalable under typical use; training and prediction are practical with standard resources or well-supported parallelization.
Interpretability (IN)	Model logic is difficult to inspect directly because of large model size, complex transformations, kernels, or opaque aggregation.	Relational features remain interpretable, but aggregation, weighting, ensemble structure, or preprocessing makes the full model less transparent.	Decision logic is compact and directly auditable, e.g. a small set of pairwise rules, pathway-level rank-disruption scores, or a readable tree structure.
Availability (AV)	No public implementation identified, code available only on request, or implementation insufficiently documented for routine reuse.	Public or author-provided implementation exists but may have limited documentation, maintenance, dependencies, or restricted usability.	Accessible software package, repository, or web tool with sufficient documentation to support independent use or reproduction.
Applicability (AP)	Narrowly specialized method; requires a specific data type, task setting, or validation scenario.	Applicable to a moderate range of related problems, but with important assumptions or modality/task restrictions.	Usable across several related relational-method settings, e.g. biomedical classification, reference-based inference, or pathway-level analysis, when assumptions are met.

Supplementary Table 13: Qualitative comparison of relational algorithms based on the rubric in Supplementary Table 12. AC: accuracy potential; SP: speed/scalability; IN: interpretability; AV: availability; AP: applicability.

Algorithm	AC	SP	IN	AV	AP
Basic RXA Algorithms					
Top Scoring Pair (TSP)	*	***	***	***	***
k-Top Scoring Pair (k-TSP)	**	**	**	***	***
Weighted TSP (WTSP)	**	**	**	*	**
Top Scoring Triplets (TST)	**	**	**	***	**
Top Scoring N-tuples (TSN)	**	*	*	***	**
Extensions of k-TSP					
Weighted k-Top Scoring Pairs (WT-k-TSP)	**	**	**	**	**
k-GeneTriple	**	*	**	*	**
k-Top Scoring N-tuples (k-TSN)	**	**	*	*	**
Support Vector Machine with k-TSP (SVM+k-TSP)	**	**	*	***	**
Entropy-based Improved k-TSP (Ik-TSP)	**	*	**	*	**
Chi-square Statistic-based Top-Scoring Gene (Chi-TSG)	**	**	***	*	**
Maximal Information Coefficient (MIC) Method	**	**	**	*	**
Vertical and Horizontal k-TSP (VH-k-TSP)	**	*	**	*	***
AUC-based TSP (AUCTSP)	**	**	***	*	**
Linear Combination k-TSP (LC-k-TSP)	**	*	**	*	**
Kernel Function k-TSP (KF-k-TSP)	**	*	*	*	**
Evolutionary and Decision Tree-Based Algorithms					
Top-Scoring Pair Decision Tree (TSPDT)	**	*	***	***	**
Global TSP Decision Tree (GTSPDT)	**	*	***	*	**
Evolutionary TSP (EvoTSP)	**	**	***	*	**
Top Inter-Gene Relations (TIGER)	**	**	***	*	**
Relative Evolutionary Hierarchical Analysis (REHA)	**	*	**	*	**
Evolutionary Relative Expression Decision Tree (Evo-REDT)	**	**	**	*	**
Dynamic TSP on Variance (DTSP-V)	**	*	***	*	**
Advanced Relative Expression Analysis (ARXA)	**	**	***	**	**
Evolutionary Heterogeneous Decision Tree (EvoHDTree)	**	**	**	*	**
Relative Multi-test Classification Tree (RMCT)	**	**	**	*	**
Random Rank Forest (RRF)	**	**	*	***	**
Evolutionary Multi-Test Tree with RXA (EMTTree+RX)	**	*	**	*	**
Reference-Based REO Methods					
Rank-based Comparative analysis (RankComp)	**	*	***	**	***
individualized Pairwise Analysis of Gene Expression (iPAGE)	**	*	***	**	***
Specialized and Hybrid Algorithms					
Relative Expression Analysis with Gene-set Pairs (RXA-GSP)	**	**	**	**	**
Bayesian TSP (BTSP)	**	*	**	***	***
Dynamic TSP on Variance (DTSP-V)	**	*	***	*	**
KNN+Relative Relation Metric (KNN+RRM)	**	**	*	***	**
Pairwise SVM ensemble (PSVM)	**	*	*	*	**
multi-omics enhanced Gene Pair Signature (meGPS)	**	*	**	*	**
Platform Independent One-Sample classifier (PIANOS)	**	*	**	**	**

Supplementary Table 14: Detailed evaluation of Top Scoring Pair (TSP) Geman et al. (2004)

Criterion	Rating	Justification
Accuracy	∗	TSP relies on a single gene pair for classification. In the original study it reported useful performance across several datasets, but it can underperform when multiple independent relations contribute to class separation. Its accuracy potential is therefore rated conservatively.
Speed	∗ ∗ ∗	TSP is computationally simple: training evaluates candidate gene pairs and prediction uses one retained pair. The dominant pair-scoring cost is $O(P^2 \cdot M)$, which is usually practical after standard feature filtering.
Interpretability	∗ ∗ ∗	The model is directly auditable because the final decision is based on one relation, such as $x_i > x_j$. This makes the rule easy to inspect, although biological interpretation still requires external validation.
Availability	∗ ∗ ∗	TSP is available in established tools, including R packages such as `tspair`. Its simple algorithmic form also makes independent implementation straightforward.
Applicability	∗ ∗ ∗	TSP has been widely used as a baseline and compact classifier for binary gene-expression classification tasks. Its applicability is strongest when a small number of stable ordering reversals is expected.

General Comment: TSP is a foundational RXA algorithm valued for simplicity, speed, and interpretability. It is useful for quick analyses and as a baseline, but its reliance on a single gene pair can limit performance in datasets where multiple relations contribute to class distinctions.

Supplementary Table 15: Detailed evaluation of k-Top Scoring Pair (k-TSP) Tan et al. (2005)

Criterion	Rating	Justification
Accuracy	∗∗	k-TSP extends TSP by combining multiple disjoint gene pairs through voting. The original study reported higher average accuracy than single-pair TSP and performance comparable to SVMs in the evaluated binary tasks, but performance remains dataset-dependent.
Speed	∗∗	Pair scoring retains the $O(P^2 \cdot M)$ structure of TSP, but selecting k and evaluating multiple pairs adds model-selection cost. The method remains practical for many gene-expression datasets, especially after feature filtering.
Interpretability	∗∗	k-TSP remains relatively interpretable because the model consists of a limited set of pairwise rules. Interpretability decreases as the number of selected pairs grows or when voting behaviour becomes less transparent.
Availability	∗ ∗ ∗	k-TSP is implemented in accessible tools, most notably the `switchBox` R package. This supports independent use and reproducible benchmarking.
Applicability	∗ ∗ ∗	k-TSP is broadly usable for binary gene-expression classification and can be adapted to multi-class problems through decomposition strategies. Its assumptions and performance should still be checked for each dataset.

General Comment: k-TSP provides a practical extension of TSP by using multiple gene pairs while retaining a compact voting structure. It is a useful baseline for gene-expression classification tasks, although its performance and stability remain dataset-dependent.

Supplementary Table 16: Detailed evaluation of Weighted Top Scoring Pair (WTSP) Luo et al. (2008)

Criterion	Rating	Justification
Accuracy	∗∗	WTSP modifies TSP scoring by incorporating class priors and misclassification costs. The original study reported improved accuracy over TSP in selected imbalanced settings, including the GCM example, but the evidence is specific to the evaluated datasets.
Speed	∗∗	WTSP adds scoring components relative to TSP. The reported accelerated cross-validation scheme can reduce the number of candidate pairs considered, but implementation details affect practical runtime.
Interpretability	∗∗	WTSP retains the pairwise structure of TSP, but class priors and misclassification costs make the scoring rule less minimal than the original single-pair classifier.
Availability	∗	No ready-to-use public software package was identified in the original publication. Reuse therefore likely requires reimplementation from the paper.
Applicability	∗∗	WTSP is most relevant when class imbalance or unequal misclassification costs are important. It is more specialized than TSP or k-TSP.

General Comment: WTSP addresses class imbalance within the TSP framework while retaining a compact pairwise rule structure. Its main practical limitation is the lack of a readily available implementation.

Supplementary Table 17: Detailed evaluation of Top-Scoring Triplets (TST) Lin et al. (2009)

Criterion	Rating	Justification
Accuracy	**	TST extends TSP from pairs to triplets. The original study reported improved performance over TSP in several evaluated cancer datasets, including BRCA1 mutation-status prediction, but the gain over multi-pair methods is not necessarily consistent across settings.
Speed	**	Exhaustive triplet search increases the candidate space from $O(P^3)$. Practical use therefore depends on filtering, pruning, or other search-space reduction strategies.
Interpretability	**	TST remains rule-based, but triplet orderings are less transparent than pairwise switches. Interpretation is possible, but usually requires more effort than for TSP or k-TSP.
Availability	***	The method was made available through an R package for relative expression analysis. This improves accessibility compared with methods that require reimplementation.
Applicability	**	TST is relevant when higher-order ordering relations are plausible and the feature space can be constrained enough for practical search. Its computational demands limit use on very large unfiltered datasets.

General Comment: TST illustrates the trade-off between greater relational expressiveness and increased computational and interpretive complexity. It can be useful when triplet-level ordering patterns are plausible, but it should not be assumed to improve over pair-based models in all datasets.

Supplementary Table 18: Detailed evaluation of Top-Scoring N-tuples (TSN) Magis and Price (2012)

Criterion	Rating	Justification
Accuracy	**	TSN generalizes TSP and TST to higher-order gene orderings. The original study reported competitive performance on MAQC-II datasets, but the evidence is tied to the evaluated settings and does not establish general superiority over lower-order models.
Speed	*	The candidate space grows rapidly with tuple size. Even with GPU or CPU implementations, resource requirements can become substantial as N increases.
Interpretability	*	Interpretability decreases as tuple size grows. A specific ordering of many genes is harder to inspect and biologically contextualize than a pairwise or triplet rule.
Availability	***	CPU and GPU source-code implementations were reported for MATLAB. The need for a specific environment and, in some cases, specialized hardware may still limit routine use.
Applicability	**	TSN is applicable to gene-expression classification settings where higher-order relations are hypothesized and computational resources are available. It is less suitable as a default method for routine high-dimensional screening.

General Comment: TSN is a higher-order extension of the TSP/TST family. It increases representational capacity, but this comes with greater computational cost and reduced direct interpretability. Its use is best justified when higher-order ordering structure is part of the research question.

Supplementary Table 19: Detailed evaluation of Weighted k-Top Scoring Pairs (WT-k-TSP) Czajkowski and Kretowski (2008)

Criterion	Rating	Justification
Accuracy	**	WT-k-TSP extends k-TSP by introducing weights based on expression differences. The original study reported higher accuracy than k-TSP in several evaluated microarray datasets, but the evidence remains tied to the tested datasets and protocol.
Speed	**	The method retains the pairwise search structure of k-TSP, with additional overhead from weighting. Reported computation times were comparable to k-TSP in the original experiments, but practical runtime depends on implementation and model-selection settings.
Interpretability	**	The model remains based on pairwise comparisons, but the weighting mechanism makes the decision rule less minimal than standard k-TSP. The selected relations remain inspectable.
Availability	**	The original publication describes an implementation using specific software, but no standalone public package was identified. A partial practical analogue can be constructed in ITree Sokołowski et al. (2024), although this does not reproduce the full original implementation.
Applicability	**	WT-k-TSP is relevant when strict binary ordering may be too coarse and expression differences between paired genes are informative. Its use should be validated for the target dataset and preprocessing pipeline.

General Comment: WT-k-TSP is a weighted extension of k-TSP that retains pairwise interpretability while adding a scale-dependent component. Its main practical limitation is the lack of a standard public implementation.

Supplementary Table 20: Detailed evaluation of k-GeneTriple Yoon et al. (2008)

Criterion	Rating	Justification
Accuracy	**	k-GeneTriple extends k-TSP from pairs to triplets. The original study reported favourable results on prostate and colon cancer datasets, especially when datasets were integrated, but these results should be interpreted as study-specific evidence rather than general superiority.
Speed	*	Triplet search increases the candidate space to $O(P^3)$. The reported pre-selection of informative genes can reduce runtime, but this introduces an additional filtering step that must be handled carefully in predictive validation.
Interpretability	**	The method produces explicit triplet-based ordering rules. These remain auditable, but interpretation is less direct than for pairwise TSP or k-TSP rules.
Availability	*	No public ready-to-use software package or source code was identified from the publication. Independent reuse likely requires reimplementation.
Applicability	**	The method is relevant when higher-order ordering relations are hypothesized and the feature space can be constrained. Its applicability is limited by computational cost and software availability.

General Comment: k-GeneTriple is a triplet-based extension of k-TSP. It increases representational capacity but also increases computational and interpretive complexity.

Supplementary Table 21: Detailed evaluation of k-Top Scoring N-tuples (k-TSN) Yoon et al. (2010)

Criterion	Rating	Justification
Accuracy	**	k-TSN generalizes k-TSP to N-tuples. The original study reported favourable performance on prostate and colon cancer datasets, but the evidence is specific to the evaluated settings and does not justify a universal high-accuracy claim.
Speed	**	The method uses a strategy that combines shorter rules into longer ones and prunes the search space. This can reduce runtime compared with exhaustive N-tuple search, although complexity still increases with tuple size and validation design.
Interpretability	*	Although the rules are explicit, interpretability decreases as tuple size increases. Long ordered tuples are harder to inspect and biologically contextualize than pairwise or triplet relations.
Availability	*	No public ready-to-use software package or source code was identified. This limits independent reuse and benchmarking.
Applicability	**	k-TSN is relevant for problems where higher-order ordering relations are plausible and sufficient data are available to estimate them. It is less suitable as a default method for routine high-dimensional screening.

General Comment: k-TSN extends the k-TSP paradigm to higher-order relations while attempting to reduce the search burden. Its practical use is constrained by interpretability, validation needs, and software availability.

Supplementary Table 22: Detailed evaluation of Support Vector Machine with k-TSP (SVM+k-TSP) Shi et al. (2011)

Criterion	Rating	Justification
Accuracy	**	This hybrid method uses k-TSP-derived relational features within an SVM classifier. The original study reported improved performance over k-TSP and SVM alone in the evaluated cancer prognostic datasets, but the evidence remains protocol- and dataset-specific.
Speed	**	The k-TSP feature-construction step is relatively efficient, but SVM training and parameter tuning can add computational cost. Runtime depends on the number of selected relational features and the validation scheme.
Interpretability	*	The selected k-TSP features remain interpretable, but the final SVM decision boundary is less directly auditable than a small rule set or decision tree.
Availability	***	MATLAB code was reported as available from the authors' project website, with external dependencies. This improves reproducibility relative to methods without accessible code.
Applicability	**	The method is relevant when relational feature construction is useful but a more flexible decision boundary is desired. Its interpretability and performance should be assessed relative to both pure relational models and standard value-based baselines.

General Comment: SVM+k-TSP is an embedded/hybrid relational method. It can improve predictive flexibility relative to pure k-TSP, but this comes with reduced transparency of the final classifier.

Supplementary Table 23: Detailed evaluation of Entropy-based Improved k-TSP (Ik-TSP) Zhou (2012)

Criterion	Rating	Justification
Accuracy	**	Ik-TSP adds entropy-based feature pre-selection before k-TSP modelling. The original study reported favourable average accuracy across several binary gene-expression datasets, but performance varied by task and the evidence is not equivalent to a harmonized external benchmark.
Speed	*	The entropy-based pre-selection step adds computational cost before pair search. It may reduce the final search space, but it also introduces an additional data-dependent step that must be controlled during validation.
Interpretability	**	The final classifier remains based on a limited set of gene pairs, but the entropy-based pre-selection step adds a less directly interpretable statistical filtering layer.
Availability	*	No public ready-to-use software or source code was identified from the publication. This limits independent reuse.
Applicability	**	Ik-TSP is relevant for high-dimensional gene-expression datasets where feature pre-selection is desirable. Its practical value depends on leakage-free filtering and task-specific validation.

General Comment: Ik-TSP combines entropy-based filtering with k-TSP. It can reduce the candidate feature space, but its evaluation requires careful validation because the filtering step is data-dependent.

Supplementary Table 24: Detailed evaluation of Chi-square Statistic-based Top-Scoring Gene (Chi-TSG) Wang et al. (2013)

Criterion	Rating	Justification
Accuracy	**	Chi-TSG extends TSP/k-TSP by using the chi-square statistic to select marker sets and account for joint effects among multiple genes. The original study reported favourable results across binary and multi-class datasets, but the evidence remains tied to the evaluated benchmarks.
Speed	**	Computing chi-square statistics for marker pairs or sets is more demanding than simple pairwise rank-reversal scoring. However, practical constraints on the number of markers can keep the search manageable in selected gene-expression settings.
Interpretability	***	The method retains an explicit marker-set structure and uses a recognizable statistical scoring criterion. This supports auditability, although interpretation becomes less direct as the number of selected markers increases.
Availability	*	No public ready-to-use implementation was identified from the publication. Independent reuse likely requires reimplementation.
Applicability	**	Chi-TSG is relevant for binary and multi-class classification settings where marker-set effects may be useful. Its practical use is limited by software availability and the need for task-specific validation.

General Comment: Chi-TSG is a statistically motivated extension of the TSP framework. It provides an interpretable marker-set formulation, but its practical accessibility is limited by the lack of public software.

Supplementary Table 25: Detailed evaluation of Maximal Information Coefficient (MIC) Method Chen et al. (2016)

Criterion	Rating	Justification
Accuracy	**	The MIC-based method uses maximal information coefficient scoring to identify synergistic gene pairs. Reported results suggest that such pairs can be useful when combined with a classifier, but the evidence is specific to the evaluated datasets and implementation.
Speed	**	MIC calculation over many gene pairs is computationally demanding because the number of candidate pairs grows quadratically. Runtime can become limiting for large feature spaces without filtering or parallelization.
Interpretability	**	The method identifies explicit gene pairs, which supports hypothesis generation. However, the MIC score itself is less directly interpretable than a simple ordering rule.
Availability	**	General MIC estimators are available, but the specific three-variable implementation used in the study was not identified as a dedicated ready-to-use package.
Applicability	**	The method is relevant when non-linear or synergistic pairwise associations are of interest. Its use requires attention to computational cost, classifier choice, and validation design.

General Comment: The MIC-based approach is an adjacent pairwise feature-selection method rather than a standard RXA rule classifier. It can support discovery of candidate interactions, but routine use is constrained by computation and implementation availability.

Supplementary Table 26: Detailed evaluation of Vertical and Horizontal k-TSP (VH-k-TSP) Huang et al. (2018)

Criterion	Rating	Justification
Accuracy	**	VH-k-TSP combines horizontal within-sample comparisons with vertical cross-sample relationships. The original study reported favourable results relative to standard k-TSP in several evaluated datasets, but these findings remain dataset- and protocol-specific.
Speed	*	Evaluating both vertical and horizontal relations increases computational complexity relative to standard k-TSP. This can limit scalability in very high-dimensional datasets.
Interpretability	**	The method still uses explicit comparisons, but the combination of vertical and horizontal relation types makes interpretation less direct than in standard TSP or k-TSP.
Availability	*	No public implementation was identified from the publication. Reuse likely requires reimplementation from the methodological description.
Applicability	**	VH-k-TSP is relevant when both within-sample and cross-sample relational patterns are of interest. Its applicability should be assessed separately for each data modality and validation setting.

General Comment: VH-k-TSP extends k-TSP by incorporating an additional cross-sample relational component. This increases representational flexibility, but also increases computational and interpretive complexity.

Supplementary Table 27: Detailed evaluation of AUC-based Top Scoring Pair (AUCTSP) Kagaris et al. (2018)

Criterion	Rating	Justification
Accuracy	**	AUCTSP modifies TSP scoring by using the area under the ROC curve to evaluate gene pairs. The original study reported improved results over standard TSP in several datasets, but the evidence remains specific to the evaluated settings.
Speed	**	Computing an AUC-based score for all gene pairs adds overhead relative to the original TSP score. Because the method remains pairwise, it is still practical after appropriate feature filtering.
Interpretability	***	AUCTSP preserves the pairwise rule structure of TSP while replacing the scoring function. The selected relation remains directly auditable, although the AUC-based selection criterion is more statistical than the original reversal-count logic.
Availability	*	The publication mentions an implementation but no public ready-to-use software package was identified.
Applicability	**	AUCTSP is relevant when a more discriminative pair-scoring criterion is desired while retaining the compact TSP representation. Its use should be validated against simpler TSP/k-TSP baselines.

General Comment: AUCTSP is a scoring-level extension of TSP. It keeps the model compact and interpretable, but its practical reuse is limited by software availability.

Supplementary Table 28: Detailed evaluation of Linear Combination k-Top Scoring Pairs (LC-k-TSP) Lin et al. (2019)

Criterion	Rating	Justification
Accuracy	**	LC-k-TSP replaces a simple pairwise ordering rule with a learned linear separator for each pair. The original study reported favourable performance relative to k-TSP in several public datasets, but these findings are not equivalent to a harmonized independent benchmark.
Speed	*	Learning a linear separator for many candidate pairs is computationally demanding. Filtering can reduce the search space, but the method remains slower than standard k-TSP.
Interpretability	**	Each selected relation can be visualized as a two-feature linear boundary, but the rule is less transparent than $x_i > x_j$. Interpretability therefore depends on the number of selected pairs and the complexity of the learned boundaries.
Availability	*	No dedicated public implementation of LC-k-TSP was identified. The reported implementation context does not provide the same accessibility as maintained package software.
Applicability	**	LC-k-TSP is relevant when strict order comparisons are too restrictive and pairwise linear boundaries are plausible. Its practical use is constrained by computation and software availability.

General Comment: LC-k-TSP increases flexibility relative to standard k-TSP by learning pairwise linear boundaries. This can improve fit in selected settings, but at the cost of higher computation and reduced simplicity.

Supplementary Table 29: Detailed evaluation of Kernel Function k-Top Scoring Pairs (KF-k-TSP) Li et al. (2023)

Criterion	Rating	Justification
Accuracy	**	KF-k-TSP extends k-TSP with kernel functions to model non-linear pairwise relations. The original study reported favourable performance compared with several TSP-family methods, but the evidence remains specific to the reported datasets and tuning choices.
Speed	*	Kernel-based pair evaluation increases computational cost relative to standard k-TSP, especially when multiple kernels or extensive tuning are used.
Interpretability	*	Although the method is still organized around feature pairs, kernelized decision boundaries are less directly auditable than simple ordering rules.
Availability	*	No public ready-to-use software or code was identified from the publication, limiting independent reuse and comparison.
Applicability	**	KF-k-TSP is relevant for research settings where non-linear pairwise effects are plausible and predictive flexibility is prioritized over direct rule interpretability.

General Comment: KF-k-TSP increases the flexibility of k-TSP through kernelized pairwise modelling. This reduces direct interpretability and increases computational burden, so its use should be justified by the task and validation evidence.

Supplementary Table 30: Detailed evaluation of Top-Scoring Pair Decision Tree (TSPDT) Czajkowski and Kretowski (2011)

Criterion	Rating	Justification
Accuracy	**	TSPDT integrates TSP logic into a decision-tree structure. Reported experiments showed improvements over single-pair TSP in selected datasets, but the greedy top-down induction can lead to suboptimal splits or overfitting.
Speed	*	At each node, TSPDT searches candidate gene pairs, which can be computationally demanding in high-dimensional data. Training-split-only feature filtering can reduce cost, but the method is slower than simple TSP/k-TSP or standard univariate decision trees.
Interpretability	***	The final model is a tree of explicit gene-pair comparisons, which can be inspected as a flowchart. Biological interpretation of selected pairs still requires external evidence.
Availability	***	Although the original publication did not provide a modern ready-to-use package, an R implementation is available through *BigTSP*, and related single-test tree construction can be explored using ITree Sokołowski et al. (2024).
Applicability	**	TSPDT is relevant for binary gene-expression classification when hierarchical rule structure is desirable. Its practical use depends on feature filtering, validation design, and software compatibility.

General Comment: TSPDT is a hierarchical RXA method that combines pairwise rules with decision-tree structure. Its strengths are auditability and compact rule representation; its limitations are greedy induction and computational cost.

Supplementary Table 31: Detailed evaluation of Global Top-Scoring Pair Decision Tree (GTSPDT) Czajkowski and Kretowski (2013)

Criterion	Rating	Justification
Accuracy	**	GTSPDT uses an evolutionary algorithm to optimize TSP-based tree structure more globally than greedy top-down induction. Reported results were favourable relative to selected RXA baselines, but the evidence remains study-specific.
Speed	*	Evolutionary optimization of tree structure is computationally demanding and slower than greedy induction. Runtime depends on population size, number of generations, fitness evaluation cost, and stopping criteria.
Interpretability	***	The final output remains a decision tree with explicit gene-pair tests. This supports model inspection, although the evolutionary search process itself is less transparent than greedy induction.
Availability	*	No public ready-to-use implementation was identified. Independent reuse likely requires reimplementation.
Applicability	**	GTSPDT is relevant when a hierarchical RXA model is desired and computational budget allows global or near-global tree search. Its use is constrained by runtime and software availability.

General Comment: GTSPDT extends TSPDT by replacing greedy induction with evolutionary structure optimization. This increases search flexibility but also increases computational cost and reproducibility requirements.

Supplementary Table 32: Detailed evaluation of Evolutionary Top-Scoring Pair (EvoTSP) Czajkowski and Kretowski (2014)

Criterion	Rating	Justification
Accuracy	**	EvoTSP uses evolutionary search to select and weight gene pairs. Reported experiments showed favourable performance relative to several RXA baselines, but these results depend on the evaluated datasets, fitness function, and evolutionary settings.
Speed	**	EvoTSP is more computationally demanding than deterministic k-TSP, but its flat representation is simpler than evolutionary tree models. Runtime depends on population size, number of generations, and fitness-evaluation cost.
Interpretability	***	The final classifier consists of a set of weighted gene-pair relations. The selected relations remain inspectable, although weights add a modest layer of complexity.
Availability	*	No public ready-to-use implementation was identified. This limits independent reuse and comparative benchmarking.
Applicability	**	EvoTSP is relevant for binary gene-expression classification when weighted sets of relations are desirable. It requires careful tuning and repeated-run assessment because of stochastic search.

General Comment: EvoTSP is an evolutionary extension of k-TSP that searches for weighted relation sets. It retains a relatively compact representation, but practical use is limited by stochastic optimization and lack of public software.

Supplementary Table 33: Detailed evaluation of Top Inter-GEne Relations (TIGER) Czajkowski et al. (2016)

Criterion	Rating	Justification
Accuracy	**	TIGER uses evolutionary search to identify weighted gene relations, including pairs and triplets. The original study reported favourable results relative to selected RXA baselines, but performance depends on the dataset, relation family, and evolutionary settings.
Speed	**	TIGER is more computationally intensive than simple exhaustive pairwise methods, but uses a compact horizontal representation and search heuristics to reduce the burden. Runtime remains dependent on implementation and stopping criteria.
Interpretability	***	The final model consists of a relatively small set of weighted relations. These relations are inspectable, although interpretation is less direct for triplets or weighted combinations than for single-pair rules.
Availability	*	No public ready-to-use software or code was identified. This limits independent validation and routine use.
Applicability	**	TIGER is relevant when pairwise and triplet relations are of interest and a stochastic search over relation sets is acceptable. Its use requires careful validation and parameter reporting.

General Comment: TIGER extends horizontal RXA models by allowing evolutionary search over weighted relations. It increases expressiveness while retaining explicit relation outputs, but reproducibility and accessibility are limited by the lack of public implementation.

Supplementary Table 34: Detailed evaluation of Relative Evolutionary Hierarchical Analysis (REHA) Czajkowski and Kretowski (2019)

Criterion	Rating	Justification
Accuracy	**	REHA uses evolutionary search to construct hierarchical gene-cluster decision structures. Reported experiments showed favourable results relative to several RXA baselines, but the evidence remains dataset-specific and should not be interpreted as general superiority.
Speed	*	The combination of evolutionary search and hierarchical gene grouping is computationally demanding. Training time is typically higher than for simpler RXA or greedy tree models.
Interpretability	**	The final model is tree-structured, but internal gene clusters make interpretation more complex than in TSPDT or other simple gene-pair trees.
Availability	*	No public ready-to-use implementation was identified. This limits reproducibility and practical adoption.
Applicability	**	REHA is relevant for gene-expression classification tasks where hierarchical gene-group structure is of interest. Its use is constrained by computational cost, software availability, and validation requirements.

General Comment: REHA is an evolutionary hierarchical model that introduces gene clusters into RXA-style tree induction. It may generate useful biological hypotheses, but it is computationally demanding and lacks public software.

Supplementary Table 35: Detailed evaluation of Evolutionary Relative Expression Decision Tree (Evo-REDT) Czajkowski et al. (2020a)

Criterion	Rating	Justification
Accuracy	**	Evo-REDT induces decision trees using weighted gene-pair tests and evolutionary search. Reported experiments showed favourable results compared with selected RXA baselines, but performance claims remain tied to the evaluated datasets and settings.
Speed	**	Evo-REDT is computationally intensive, but reported OpenMP and GPU acceleration reduced runtime in the original implementation. The speed rating assumes availability of suitable parallel hardware.
Interpretability	**	The output is a decision tree with weighted gene-pair tests. This is more auditable than many black-box learners, but less minimal than TSP or TSPDT because relation weights and evolved structure add complexity.
Availability	*	No public ready-to-use implementation was identified. This limits independent reuse and benchmarking.
Applicability	**	Evo-REDT is relevant for gene-expression classification tasks where weighted relations and tree structure are useful. Its applicability depends on computational resources and careful model-selection design.

General Comment: Evo-REDT is an evolutionary decision-tree model based on weighted gene-pair tests. It increases modelling flexibility relative to simpler RXA trees, but requires substantial computation and is not publicly available as reusable software.

Supplementary Table 36: Detailed evaluation of Relative eXpression Classification Tree (RXCT) Czajkowski et al. (2020b)

Criterion	Rating	Justification
Accuracy	**	RXCT uses top-down tree induction with pairwise or triplet relational tests. Reported experiments showed favourable performance relative to several RXA baselines, but these findings are specific to the evaluated datasets and protocol.
Speed	**	RXCT relies on GPU acceleration to make node-level pair or triplet search practical. This can substantially reduce runtime, but speed depends on hardware availability, implementation, and feature filtering.
Interpretability	**	RXCT produces decision trees with explicit relational tests. Interpretability is retained to some extent, although triplet tests and larger trees are less directly auditable than single-pair rules.
Availability	*	No public ready-to-use implementation was identified. This limits accessibility and independent comparison.
Applicability	**	RXCT is relevant for classification tasks where pairwise or triplet relational splits are desired and GPU resources are available. It requires careful validation because the search space can be large.

General Comment: RXCT is a top-down relational decision-tree method that uses GPU acceleration for node-level relational search. It provides an auditable tree output, but practical reuse is limited by hardware needs and lack of public software.

Supplementary Table 37: Detailed evaluation of Advanced Relative Expression Analysis (ARXA) Czajkowski et al. (2020c)

Criterion	Rating	Justification
Accuracy	**	ARXA introduces a ratio-based scoring mechanism within a decision-tree framework. Reported experiments showed favourable results relative to selected RXA baselines, but these results remain dataset- and protocol-specific.
Speed	**	ARXA uses GPU acceleration to score many candidate relations. Runtime can be practical when suitable hardware is available, but the method remains dependent on implementation and feature-space size.
Interpretability	***	The final trees use ratio-based tests and thresholds, which can be inspected directly. Interpretation is still conditional on the biological plausibility and validation of selected relations.
Availability	**	No full public implementation was identified. Some related single-test logic can be explored through ITree Sokołowski et al. (2024), but this does not reproduce the complete ARXA algorithm.
Applicability	**	ARXA is relevant when ratio-based relations are plausible and tree-based auditability is desired. Its use depends on hardware availability, implementation access, and validation design.

General Comment: ARXA is a ratio-based relational tree method. It offers an interpretable tree output, but its practical use is limited by incomplete public availability and dependence on specialized implementation.

Supplementary Table 38: Detailed evaluation of Evolutionary Heterogeneous Decision Tree (EvoHDTree) Czajkowski et al. (2021)

Criterion	Rating	Justification
Accuracy	**	EvoHDTree uses evolutionary search to build heterogeneous decision trees in which nodes can contain different test types, including univariate, TSP-like, and weighted TSP-like tests. Reported experiments showed favourable performance in the evaluated settings, but the evidence remains dataset- and protocol-specific.
Speed	**	The method is computationally intensive because of evolutionary optimization, but the reported implementation uses OpenMP and GPU acceleration to reduce runtime. Practical speed depends on hardware, population size, stopping criteria, and feature-space size.
Interpretability	**	The final output is a decision tree, but heterogeneous node types make interpretation less direct than in simple TSPDT-like models. Individual tests remain inspectable.
Availability	*	No public ready-to-use implementation was identified. This limits independent reuse and benchmarking.
Applicability	**	EvoHDTree is relevant for classification settings where heterogeneous test types may be useful. Its application requires careful tuning, validation, and access to suitable computational resources.

General Comment: EvoHDTree extends relational tree modelling by allowing heterogeneous test types within an evolutionary framework. This increases modelling flexibility, but also increases computational and validation requirements.

Supplementary Table 39: Detailed evaluation of Relative Multi-test Classification Tree (RMCT) Czajkowski et al. (2025a)

Criterion	Rating	Justification
Accuracy	**	RMCT reported competitive performance across the evaluated multi-omics datasets, including comparisons with standard decision-tree and ensemble baselines. By using "multi-tests" composed of several TSP rules at each node, it is designed to capture ordering relationships across defined feature groups. The evidence remains method- and dataset-specific.
Speed	**	Searching for multi-tests is computationally demanding, but the reported implementation uses GPU acceleration. The speed rating assumes access to suitable hardware and does not imply low computational cost in all settings.
Interpretability	**	RMCT retains a decision-tree structure, but each split may combine several TSP rules. This makes the model more auditable than many ensemble or black-box models, but less transparent than a single-pair rule or simple TSPDT split.
Availability	*	The publication describes the algorithm but no public ready-to-use implementation was identified. This limits independent reuse and comparison.
Applicability	**	RMCT is relevant for selected multi-omics classification tasks, especially when relations can be defined within or across well-specified omics layers. Its applicability depends on feature comparability, validation design, and sufficient training data.

General Comment: RMCT is a top-down relational decision-tree method that extends single-test splits to multi-test nodes. Its practical limitations include computational requirements, task-specific validation needs, and lack of a public implementation.

Supplementary Table 40: Detailed evaluation of Evolutionary Multi-Test Tree with Relative Expression (EMTTree+RX) Czajkowski et al. (2025b)

Criterion	Rating	Justification
Accuracy	**	EMTTree+RX reported competitive performance on the evaluated multi-omics datasets. By combining evolutionary search for global tree structure with multi-test nodes containing univariate and TSP-based tests, it can explore more complex structures than greedy single-test trees. The available evidence remains dataset-specific and does not establish general superiority over other RXA or standard ML methods.
Speed	*	Global evolutionary optimization of both tree structure and local multi-test composition is computationally expensive. Runtime depends on population size, number of generations, fitness-evaluation cost, and stopping criteria.
Interpretability	**	The final output is a decision tree with multi-test nodes, so the model remains more auditable than many black-box learners. However, interpretability is lower than for single-pair TSP or simple TSPDT models because each node may combine multiple tests and test types.
Availability	*	No public ready-to-use implementation was identified. This limits independent reuse and routine benchmarking.
Applicability	**	EMTTree+RX is relevant for selected multi-omics classification tasks where heterogeneous test types and tree-based auditability are desirable. Its applicability is constrained by computational cost, tuning requirements, and the need for careful validation.

General Comment: EMTTree+RX is an evolutionary relational decision-tree model that combines global tree optimization with flexible multi-test nodes. It is methodologically expressive, but this comes at high computational cost and requires careful validation. Its practical impact is limited by the lack of a public implementation.

Supplementary Table 41: Detailed evaluation of Random Rank Forest (RRF) Lu et al. (2024)

Criterion	Rating	Justification
Accuracy	**	RRF extends rank-based decision trees into an ensemble framework. Reported performance should be treated as implementation- and dataset-dependent. In the harmonized benchmark reported in the main manuscript, the executable `ranktreeEnsemble::rforest` wrapper produced the weakest non-majority aggregate result under the common 200-feature protocol.
Speed	**	RRF is more computationally demanding than a single tree, but individual trees can be trained in parallel. Its speed is therefore moderate and depends on the implementation and ensemble size.
Interpretability	*	Individual rank-based trees may be interpretable, but the final prediction aggregates many trees. The overall model is therefore less directly auditable than a single rule, a small voting rule set, or a compact decision tree.
Availability	***	RRF is accessible through the `ranktreeEnsemble` R package on CRAN. This availability supports independent use and benchmarking.
Applicability	**	RRF is relevant for gene-expression classification tasks where a rank-based ensemble is desired. However, the benchmark in this review does not support treating the tested wrapper as a default high-performance option under the evaluated protocol.

General Comment: RRF is useful as an accessible rank-based ensemble implementation and is therefore important from a reproducibility perspective. Its predictive value should be assessed empirically for each task, and the weak aggregate result of the tested wrapper in this benchmark should be reported transparently.

Supplementary Table 42: Detailed evaluation of Relative Expression Analysis with Gene-set Pairs (RXA-GSP) Yang et al. (2013)

Criterion	Rating	Justification
Accuracy	**	RXA-GSP evaluates pairs of predefined gene sets for prognostic or classification tasks. The original study reported useful stratification in breast cancer settings, but the evidence remains tied to the evaluated cohorts, endpoint, and gene-set definitions.
Speed	**	The method evaluates many gene-set pairs, which can be computationally demanding. The search is more tractable than unrestricted gene-pair search when the number of predefined gene sets is limited.
Interpretability	**	RXA-GSP operates at the gene-set or pathway level, which can support biological interpretation. However, pathway-level summaries may hide heterogeneity among individual genes.
Availability	**	The method was implemented within the AUREA software system. This provides some practical accessibility, but it is not available as a standalone broadly maintained package.
Applicability	**	RXA-GSP is most relevant when meaningful predefined gene sets are available and pathway-level hypotheses are central to the analysis. It is less suitable when no reliable feature grouping exists.

General Comment: RXA-GSP is a pathway-oriented extension of RXA. Its main value is the ability to test gene-set-level relational hypotheses, while its limitations include dependence on predefined gene sets and limited standalone availability.

Supplementary Table 43: Detailed evaluation of Bayesian Top-Scoring Pairs (BTSP) Arslan and Braga-Neto (2017)

Criterion	Rating	Justification
Accuracy	**	BTSP extends TSP using Bayesian inference and a Bradley–Terry formulation. Reported experiments suggested favourable performance relative to standard TSP variants in selected gene-expression datasets, but the evidence remains dataset- and modelling-assumption-specific.
Speed	*	The Bayesian formulation introduces additional computational cost, including sampling or iterative estimation. This makes BTSP slower than deterministic TSP or k-TSP methods, especially for large feature spaces.
Interpretability	**	The final model remains based on gene-pair rankings, but the Bayesian scoring layer is less transparent than simple rank-reversal counting. The probabilistic formulation may aid uncertainty-aware interpretation.
Availability	***	BTSP is available as an R package on GitHub, which supports reuse and independent experimentation.
Applicability	***	BTSP is applicable to several TSP-like classification settings, including settings where probabilistic pair scoring is desirable. Its computational cost and modelling assumptions should be considered before routine use.

General Comment: BTSP is a Bayesian extension of the TSP framework. It adds a probabilistic modelling layer and has accessible software, but this comes with increased computational complexity relative to simpler pairwise classifiers.

Supplementary Table 44: Detailed evaluation of Dynamic Top Scoring Pairs on Variance (DTSP-V) Wu et al. (2016)

Criterion	Rating	Justification
Accuracy	**	DTSP-V adapts TSP logic to time-series data by incorporating trend and variance information. Reported experiments in toxicogenomics settings showed favourable performance, but the evidence is specific to time-series tasks and should not be generalized to static classification problems.
Speed	*	The method requires additional calculations for temporal patterns and variance-based summaries. This increases computational cost relative to standard TSP, especially for longer time series.
Interpretability	***	DTSP-V retains explicit pairwise relational logic while extending it to dynamic patterns. The selected temporal relations can be inspected, although biological interpretation depends on the time scale and experimental design.
Availability	*	No public ready-to-use implementation was identified from the publication. This limits reuse, especially given the specialized time-series setting.
Applicability	**	DTSP-V is specialized for time-series gene-expression classification. It is relevant when temporal ordering patterns are central to the biological question, but it is not a general-purpose static classifier.

General Comment: DTSP-V is a time-series extension of TSP. It is useful conceptually for dynamic relational modelling, but its practical use is constrained by task specificity, computational cost, and limited software availability.

Supplementary Table 45: Detailed evaluation of KNN+Relative Relation Metric (KNN+RRM) Kartowicz-Stolarska and Czajkowski (2024)

Criterion	Rating	Justification
Accuracy	**	KNN+RRM uses relative-relation information as a distance measure for k-NN. Reported evaluations and the benchmark in this review suggest competitive performance in selected settings, but results remain dataset-dependent and should be compared with both Euclidean k-NN and standard ML baselines.
Speed	**	The metric involves many pairwise relation comparisons, making it more expensive than simple value-space distances. Optimized implementations can reduce runtime, but cost remains relevant for large P.
Interpretability	*	The distance is based on relational agreement or disagreement between samples, which is conceptually interpretable. However, a k-NN decision based on many pairwise disagreements is less directly auditable than a small set of selected rules.
Availability	***	KNN+RRM is available as an open-source Python implementation, supporting reuse and benchmarking.
Applicability	**	KNN+RRM is relevant when a relational distance is desired for instance-based learning. It is not a standalone rule classifier and depends on the assumptions and limitations of k-NN.

General Comment: KNN+RRM embeds relational logic into distance-based classification. It is accessible and useful for comparison with standard k-NN metrics, but its interpretability and performance should be evaluated task by task.

Supplementary Table 46: Detailed evaluation of multi-omics enhanced Gene Pair Signature (meGPS) Wu et al. (2022)

Criterion	Rating	Justification
Accuracy	**	meGPS integrates gene-pair signatures across molecular layers, such as expression and methylation. The original study reported favourable prognostic results, but these should be interpreted in relation to the disease setting, omics types, cohort structure, and validation design.
Speed	*	The method searches for predictive relations within multiple omics layers and then combines them. This is computationally demanding for high-dimensional multi-omics data.
Interpretability	**	meGPS provides explicit molecular-pair signatures across omics layers, which can support hypothesis generation. Interpretation is more complex than in single-omics pairwise models because cross-layer biological relationships require additional validation.
Availability	*	No public ready-to-use implementation was identified from the original publication. This limits independent reuse and benchmarking.
Applicability	**	meGPS is relevant for selected multi-omics prognostic or classification tasks where cross-layer relational patterns are biologically plausible. Its applicability depends on feature comparability, cohort size, and validation design.

General Comment: meGPS is a multi-omics relational modelling example. It illustrates how gene-pair logic can be extended across molecular layers, but its use is constrained by computational cost, limited software availability, and the need for careful multi-omics validation.

Supplementary Table 47: Detailed evaluation of Platform Independent and Normalization-free One-Sample classifier (PIANOS) Cai et al. (2025)

Criterion	Rating	Justification
Accuracy	**	PIANOS reported favourable prognostic performance across multiple cohorts in the original disease setting. This is substantial evidence for that application, but it should not be interpreted as general performance evidence across diseases, assay platforms, or intended-use populations.
Speed	*	The framework combines pathway-level enrichment processing with training of a k-TSP ensemble. This multi-step design can be computationally demanding during model development.
Interpretability	**	PIANOS applies k-TSP-style relational logic to pathway-level scores. This supports pathway-level interpretation, but the ensemble structure is less transparent than a single rule or compact k-TSP model.
Availability	**	Public research code and example materials are available through the authors' repository/archival release, including model-construction, model-usage, and demo files. However, the framework is not a simple plug-and-play package for the common binary-classification benchmark used in Fig. 4, because reuse requires the pathway-score construction layer, method-specific dependencies, and a prognostic/survival-oriented evaluation workflow.
Applicability	**	PIANOS is relevant to supervised pathway-level prognostic modelling and one-sample-style prediction. Claims of platform independence or normalization-free operation should be interpreted as method-specific design goals that require validation under the intended assay and deployment setting.

General Comment: PIANOS is an important example of supervised k-TSP logic applied to pathway-level scores. In this review, it is best treated as a supervised pathway-level relational ensemble rather than as a task-equivalent REO or DIRAC method. Although public research code is available, its conclusions remain tied to the original colorectal-cancer prognostic setting, assay context, pathway-score preprocessing pipeline, and validation design.

References

Arslan, E., Braga-Neto, U.M.: A bayesian approach to top-scoring pairs classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 871–875. IEEE, New Orleans, LA, USA (2017). https://doi.org/10.1109/ICASSP.2017.7952280 . March 5–9, 2017. https://doi.org/10.1109/ICASSP.2017.7952280

Afsari, B., Braga-Neto, U., Geman, D.: switchbox: a Bioconductor package for calculating significant features from gene expression data. Bioinformatics **31**(1), 129–131 (2015) https://doi.org/10.1093/bioinformatics/btu564

Ambrosini, M., Gallois, C., Gandini, A., Lecanu, A., Alouani, E., Pietrantonio, F., Bergamo, F., Cremolini, C., Mazard, T., Tougeron, D., Guimbaud, R., Lonardi, S., Selves, J., Mouillet-Richard, S., Emile, J.-F., Taieb, J., Laurent-Puig, P.: From clusters to clinic: An 8-gene signature combined with mucinous component stratifies benefit of anti-ctla-4 addition to anti-pd-1 in dmmr/msi-h metastatic colorectal cancer. European Journal of Cancer, 116731 (2026) https://doi.org/10.1016/j.ejca.2026.116731

Budhraja, A., Jain, S., Jain, V.: A systematic review of feature selection techniques and their applications. Artificial Intelligence Review **56**(12), 14313–14361 (2023) https://doi.org/10.1007/s10462-022-10333-6

Chen, Y., Cao, D., Gao, J., Yuan, Z.: Discovering pair-wise synergies in microarray data. Scientific Reports **6**, 30672 (2016) https://doi.org/10.1038/srep30672

Czajkowski, M., Czajkowska, A., Kretowski, M.: TIGER: an evolutionary search for Top Inter-GEne Relations. International Journal of Data Mining and Bioinformatics **16**(2), 170–182 (2016) https://doi.org/10.1504/IJDMB.2016.080042

Czajkowski, M., Jurczuk, K., Kretowski, M.: Generic relative relations in hierarchical gene expression data classification. In: Lecture Notes in Computer Science, vol. 12270, pp. 372–384 (2020)

Czajkowski, M., Jurczuk, K., Kretowski, M.: Relative expression classification tree: A preliminary gpu-based implementation. In: Parallel Processing and Applied Mathematics (PPAM 2019). Lecture Notes in Computer Science, vol. 12043, pp. 359–369. Springer, Cham, Switzerland (2020). https://doi.org/10.1007/978-3-030-43229-4_31

Czajkowski, M., Jurczuk, K., Kretowski, M.: Tree based advanced relative expression analysis. In: Lecture Notes in Computer Science, vol. 12139, pp. 496–510 (2020)

Czajkowski, M., Jurczuk, K., Kretowski, M.: Accelerated evolutionary induction of heterogeneous decision trees for gene expression-based classification. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 946–954. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3449639.3459384

Czajkowski, M., Jurczuk, K., Kretowski, M.: Enhancing multi-omics data classification with relative expression analysis and decision trees. Journal of Computational Science **84**, 102460 (2025)

Czajkowski, M., Jurczuk, K., Kretowski, M.: Enhancing transparency of omics data analysis with the evolutionary multi-test tree and relative expression (EMTTree+RX). Expert Systems with Applications **276**, 127131 (2025)

Czajkowski, M., Kretowski, M.: Novel extension of k-tsp algorithm for microarray classification. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) New Frontiers in Applied Artificial Intelligence (IEA/AIE 2008). Lecture Notes in Computer Science, vol. 5027, pp. 456–465. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69052-8_48

Czajkowski, M., Kretowski, M.: Top Scoring Pair Decision Tree for Gene Expression Data Analysis. In: Advances in Experimental Medicine and Biology, vol. 696, pp. 27–35 (2011)

Czajkowski, M., Kretowski, M.: Global top-scoring pair decision tree for gene expression data analysis. In: Krawiec, K., Moraglio, A., Hu, T., Etaner-Uyar, A.Ş., Hu, B. (eds.) Genetic Programming. EuroGP 2013. Lecture Notes in Computer Science, vol. 7831, pp. 229–240. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37207-0_20

Czajkowski, M., Kretowski, M.: Evolutionary approach for relative gene expression algorithms. The Scientific World Journal **2014**, 593503 (2014)

Czajkowski, M., Kretowski, M.: Relative evolutionary hierarchical analysis for gene expression data classification. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '19), pp. 1156–1164. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3321707.3321862

Cai, D., Qi, H., Yang, Q., Li, H., Li, C., Hu, C., Gai, B., Zhang, X., Mao, Y., Gao, F., *et al.*: Personalized risk stratification in colorectal cancer via PIANOS system. Nature Communications **16**(1), 6561 (2025) https://doi.org/10.1038/s41467-025-61713-1

Earls, J.C., Eddy, J.A., Lank, P.M., Price, N.D.: AUREA: an open-source software system for mining relative expression in gene sets. BMC Bioinformatics **14**, 19 (2013) https://doi.org/10.1186/1471-2105-14-19

Eddy, J.A., Hood, L., Price, N.D., Geman, D.: Identifying Tightly Regulated and Variably Expressed Networks by Differential Rank Conservation (DIRAC). PLoS Computational Biology **6**(5), 1000792 (2010) https://doi.org/10.1371/journal.pcbi.1000792

Fourquet, O., Krejca, M.S., Doerr, C., Schwikowski, B.: Towards the genome-scale discovery of bivariate monotonic classifiers. BMC Bioinformatics **26**(1), 228 (2025) https://doi.org/10.1186/s12859-025-06253-7

Geman, D., d'Avignon, C., Naiman, D.Q., Winslow, R.L.: Classifying gene expression profiles from pairwise mRNA comparisons. Statistical Applications in Genetics and Molecular Biology **3**(1), 19 (2004)

Gordon, G.J., Jensen, R.V., Hsiao, L.-L., Gullans, S.R., Blumenstock, J.E., Ramaswamy, S., Richards, W.G., Sugarbaker, D.J., Bueno, R.: Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Research **62**(17), 4963–4967 (2002)

Huang, X., Chen, J., He, X.: A readily interpretable rule involving multiple forms of pairwise molecule comparisons with applications for clinical make-decision of breast cancer management. Journal of Pharmaceutical and Biomedical Analysis (2026) https://doi.org/10.1016/j.jpba.2026.117347

Huang, X., Lin, X., Zhou, L., Su, B.: Analyzing omics data by pair-wise feature evaluation with horizontal and vertical comparisons. Journal of Pharmaceutical and Biomedical Analysis **157**, 20–26 (2018) https://doi.org/10.1016/j.jpba.2018.04.052

Kim, S., Lin, C.-W., Tseng, G.C.: Metaktsp: a meta-analytic top scoring pair method for robust cross-study validation of omics prediction analysis. Bioinformatics **32**(13), 1966–1973 (2016) https://doi.org/10.1093/bioinformatics/btw115

Kartowicz-Stolarska, I.J., Czajkowski, M.: Relative relation in knn classification for gene expression data. a preliminary study. In: Marcinkowski, B., Przybylek, A., Jarzebowicz, A., Iivari, N., Insfran, E., Lang, M., Linger, H., Schneider, C. (eds.) Harnessing Opportunities: Reshaping ISD in the post-COVID-19 and Generative AI Era (ISD2024 Proceedings). University of Gdańsk, Gdańsk, Poland (2024). https://doi.org/10.62036/ISD.2024.94

Kagaris, D., Tzagarakis, G.N., Lagoudakis, M.G.: Auctsp: an improved biomarker gene pair class predictor. BMC Bioinformatics **19**(1), 244 (2018) https://doi.org/10.1186/s12859-018-2231-1

Lin, X., Afsari, B., Marchionni, L., Cope, L., Parmigiani, G., Naiman, D., Geman, D.: The ordering of expression among a few genes can provide simple cancer biomarkers and signal BRCA1 mutations. BMC Bioinformatics **10**, 256 (2009) https://doi.org/10.1186/1471-2105-10-256

Leek, J.T.: The tspair package for finding top-scoring pair classifiers in R. Bioinformatics **25**(9), 1203–1204 (2009) https://doi.org/10.1093/bioinformatics/btp128

Lin, X., Huang, X., Zhou, L., Ren, W., Zeng, J., Yao, W., Wang, X.: The robust classification model based on combinatorial features. IEEE/ACM Transactions on Computational Biology and Bioinformatics **16**(2), 650–657 (2019) https://doi.org/10.1109/TCBB.2017.2779512

Luo, Y., Liu, H., Li, J.: Weighted top score pair method for gene selection and classification **5265**, 187–198 (2008) https://doi.org/10.1007/978-3-540-87357-4_41

Liu, J., Sarker, R., Elsayed, S., Essam, D., Siswanto, N.: Large-scale evolutionary optimization: A review and comparative study. Swarm and Evolutionary Computation **85**, 101466 (2024) https://doi.org/10.1016/j.swevo.2023.101466

Li, C., Wang, T., Lin, X.: Analyzing omics data by feature combinations based on kernel functions. Journal of Bioinformatics and Computational Biology **21**(5), 2350021 (2023) https://doi.org/10.1142/S021972002350021X

Lu, M., Yin, R., Chen, X.S.: Ensemble methods of rank-based trees for single sample classification with gene expression profiles. Journal of Translational Medicine **22**, 140 (2024) https://doi.org/10.1186/s12967-024-04940-2

Lin, X., Zhang, Y., Li, C., Wang, J., Luo, P., Zhou, H.: A new data analysis method based on feature linear combination. Journal of Biomedical Informatics **94**, 103173 (2019) https://doi.org/10.1016/j.jbi.2019.103173 . Introduces LC-k-TSP for metabolomics

Mirza, B., *et al.*: Machine learning and integrative analysis of biomedical big data. Genes **10**(2), 87 (2019) https://doi.org/10.3390/genes10020087

Marzouka, N., *et al.*: multiclasspairs: an r-package for risk stratification of cancer patients using gene pair expression data. BMC Bioinformatics **22**(1), 49 (2021) https://doi.org/10.1186/s12859-021-03958-4

Molania, R., *et al.*: Prps: a pseudo-replicates pseudo-samples approach to batch correction for case-control studies. Bioinformatics **39**(1), 798 (2023) https://doi.org/10.1093/bioinformatics/btac798

Martínez-Cantín, R., *et al.*: Bayesian top-scoring pairs for robust classification of gene expression data. BMC Bioinformatics **18**(1), 375 (2017) https://doi.org/10.1186/s12859-017-1771-8

Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn. Springer, Berlin, Heidelberg (1996)

Magis, A.T., Price, N.D.: The top-scoring 'n' algorithm: a generalized relative expression classifier. BMC Bioinformatics **13**, 227 (2012) https://doi.org/10.1186/1471-2105-13-227

Qian, C., Yang, S., Chen, Y., Ge, R., Shi, F., Liu, C., Wang, H., Guo, Y.: Predicting pathological response to neoadjuvant chemoradiotherapy in locally advanced rectal cancer with two step feature selection and ensemble learning. Scientific Reports **15** (2025) https://doi.org/10.1038/s41598-025-94337-y

Richard, M., Decamps, C., Chuffart, F., Brambilla, E., Rousseaux, S., Khochbin, S., Jost, D.: PenDA, a rank-based method for personalized differential analysis: Application to lung cancer. PLoS Computational Biology **16**(5), 1007869 (2020) https://doi.org/10.1371/journal.pcbi.1007869

Sundararaman, A., *et al.*: Birch: an effective method for batch effect identification and cross-correction of heterogeneous single-cell transcriptomics data. Briefings in Bioinformatics **24**(6), 386 (2023) https://doi.org/10.1093/bib/bbad386

Sokołowski, H., Czajkowski, M., Czajkowska, A., Jurczuk, K., Kretowski, M.: ITree: a user-driven tool for interactive decision-making with classification trees. Bioinformatics **40**(5), 273 (2024) https://doi.org/10.1093/bioinformatics/btae273

Shi, P., Ray, S., Zhu, Q., Kon, M.A.: Top scoring pairs for feature selection in machine learning and applications to cancer outcome prediction. BMC Bioinformatics **12**, 375 (2011) https://doi.org/10.1186/1471-2105-12-375

Tercan, B.: Robust predictors for drug response of patients with acute myeloid leukemia. PLOS ONE (2026) https://doi.org/10.1371/journal.pone.0343422

Tian, T., Lai, G., He, M., Luo, Y., Guo, Y., Hong, G., Li, H., Song, K., Cai, H.: Exploring the influence of pre-analytical variables on gene expression measurements and relative expression orderings in cancer research. Scientific Reports **15**(1), 4489 (2025) https://doi.org/10.1038/s41598-025-88756-0

Tan, A.C., Naiman, D.Q., Xu, L., Winslow, R.L., Geman, D.: Simple decision rules for classifying human cancers from gene expression profiles. Bioinformatics **21**(20), 3896–3904 (2005)

Wang, T., *et al.*: A benchmark of batch effect correction methods for single-cell rna sequencing data. Genome Biology **24**(1), 15 (2023) https://doi.org/10.1186/s13059-022-02868-2

Wu, K., Nan, X., Chai, Y., Wang, L., Li, K.: Dtsp-v: A trend-based top scoring pairs method for classification of time series gene expression data. In: 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1787–1794 (2016). https://doi.org/10.1109/BIBM.2016.7822790

Wang, H., Sun, Q., Zhao, W., Qi, L., Gu, Y., Li, P., Zhang, M., Li, Y., Liu, S.-L., Guo, Z.: Individual-level analysis of differential expression of genes and pathways for personalized medicine. Bioinformatics **31**(1), 62–68 (2015)

Wu, C., Xie, X., Yang, X., Du, M., Lin, H., Huang, J.: Applications of gene pair methods in clinical research: advancing precision medicine. Molecular Biomedicine **6**(1), 22 (2025) https://doi.org/10.1186/s43556-025-00263-w

Wang, H., Zhang, H., Dai, Z., Chen, M.S., Yuan, Z.: Chi-TSG: a new algorithm for binary and multi-class cancer classification and informative genes selection. BMC Medical Genomics **6**(Suppl 1), 3 (2013)

Wu, Q., Zheng, X., Leung, K.-S., Wong, M.-H., Tsui, S.K.-W., Cheng, L.: megps: a multi-omics signature for hepatocellular carcinoma detection integrating methylome and transcriptome data. Bioinformatics **38**(14), 3513–3522 (2022) https://doi.org/10.1093/bioinformatics/btac379

Wang, R., Zheng, X., Wang, J., Wan, S., Song, F., Wong, M.-H., Leung, K.-S., Cheng, L.: Improving bulk rna-seq classification by transferring gene signature from single cells in acute myeloid leukemia. Briefings in Bioinformatics **23**(2) (2022) https://doi.org/10.1093/bib/bbac002

Yu, W., *et al.*: A survey of batch effects correction methods in single-cell rna sequencing. Briefings in Bioinformatics **24**(1), 569 (2023) https://doi.org/10.1093/bib/bbac569

Yoon, Y., Bien, S., Park, S.: Microarray data classifier consisting of k-top-scoring rank-comparison decision rules with a variable number of genes. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **40**(2), 216–226 (2010) https://doi.org/10.1109/TSMCC.2009.2036594

Yoon, Y., Lee, J., Park, S., Bien, S., Chung, H.C., Rha, S.Y.: Direct integration of microarrays for selecting informative genes and phenotype classification. Information Sciences **178**(1), 88–105 (2008) https://doi.org/10.1016/j.ins.2007.08.013

Yang, X., Vasudevan, P., Parekh, V., Penev, A., Cunningham, J.M.: Bridging cancer biology with the clinic: Relative expression of a GRHL2-mediated gene-set pair predicts breast cancer metastasis. PLoS ONE **8**(2), 56195 (2013) https://doi.org/10.1371/journal.pone.0056195

Yan, J., Zeng, Q., Wang, X.: Rankcompv3: a differential expression analysis algorithm based on relative expression orderings and applications in single-cell rna transcriptomics. BMC Bioinformatics **25**(1), 259 (2024) https://doi.org/10.1186/s12859-024-05889-1

Zhou, X.: An entropy-based improved k-top scoring pairs algorithm for gene expression data classification. In: 2012 International Conference on Industrial Control and Electronics Engineering, pp. 1957–1960. IEEE, Xi'an, China (2012). https://doi.org/10.1109/ICICEE.2012.533

Zheng, X., Jin, N., Wu, Q., Zhang, N., Wu, H., Wang, Y., Luo, R., Liu, T., Ding, W., Geng, Q., Cheng, L.: Less is more: relative rank is more informative than absolute abundance for compositional ngs data. Briefings in Functional Genomics **24** (2025) https://doi.org/10.1093/bfgp/elae045 . Advance article online: 2024-11-20

Zheng, G., Li, X., Wang, H.: Feature selection for machine learning in bioinformatics: A survey. Computers in Biology and Medicine **168**, 107764 (2024) https://doi.org/10.1016/j.compbiomed.2023.107764

Zheng, X., Leung, K.-S., Wong, M.-H., Cheng, L.: Long non-coding rna pairs to assist in diagnosing sepsis. BMC Genomics **22**, 286 (2021) https://doi.org/10.1186/s12864-021-07576-4

Zheng, S., Qu, W., Zhang, D., Zhou, J., Xu, Y., Wu, W., Liu, C., Huang, M., Shen, E., Chen, X., Timko, M.P., Fan, L., Yu, F., Han, D., Shen, Y.: International multicenter development of ensemble machine learning driven host response based diagnosis for tuberculosis. iScience **28**(9), 113444 (2025) https://doi.org/10.1016/j.isci.2025.113444