Network-medicine algorithms

T9 - Network-medicine-based drug repurposing

Dr. Hryhorii Chereda Prof. Dr. David B. Blumenthal

Biomedical Network Science Lab, Department AIBE, FAU Erlangen-Nürnberg

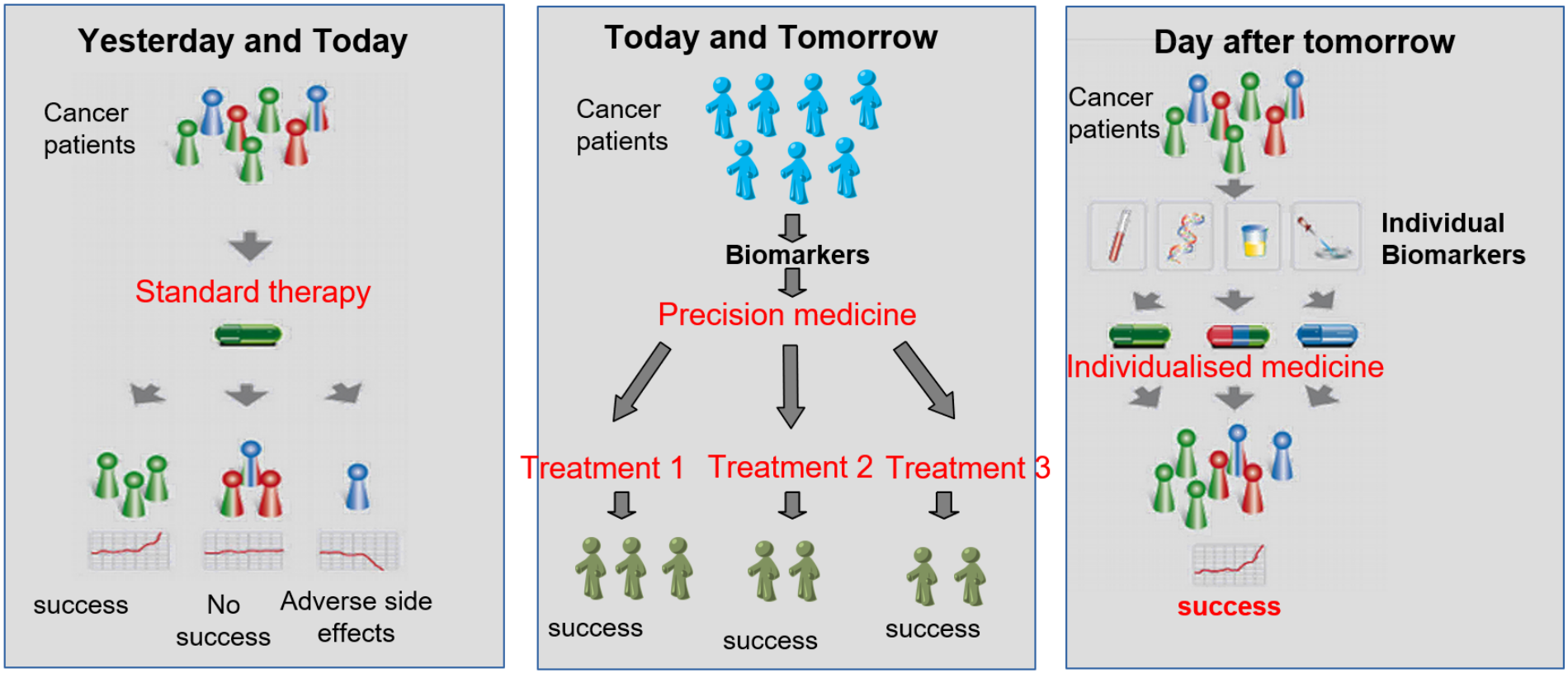
Materials adapted from Prof. Dr. David B. Blumenthal

Table of content

- Why disease module mining for precision medicine?
- Diamond and Robust
- Personalized Page Rank for drug prioritization (a.k.a. Trust Rank).

Precision medicine

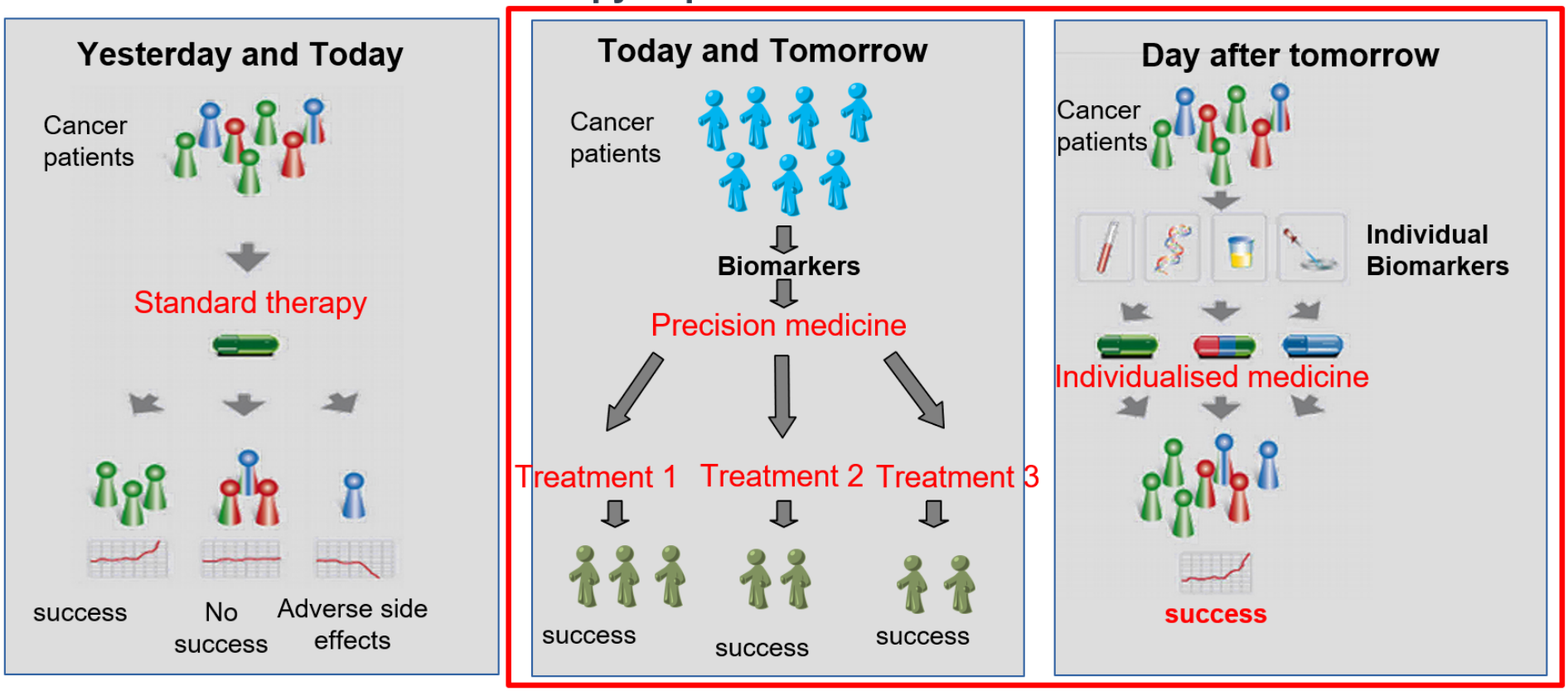
Transition from one-size-fits-all therapy to precision medicine and individualized medicine:



Courtesy Alexander König, Volker Ellenrieder UMG 2019

Precision medicine

Transition from one-size-fits-all therapy to precision medicine and individualized medicine:

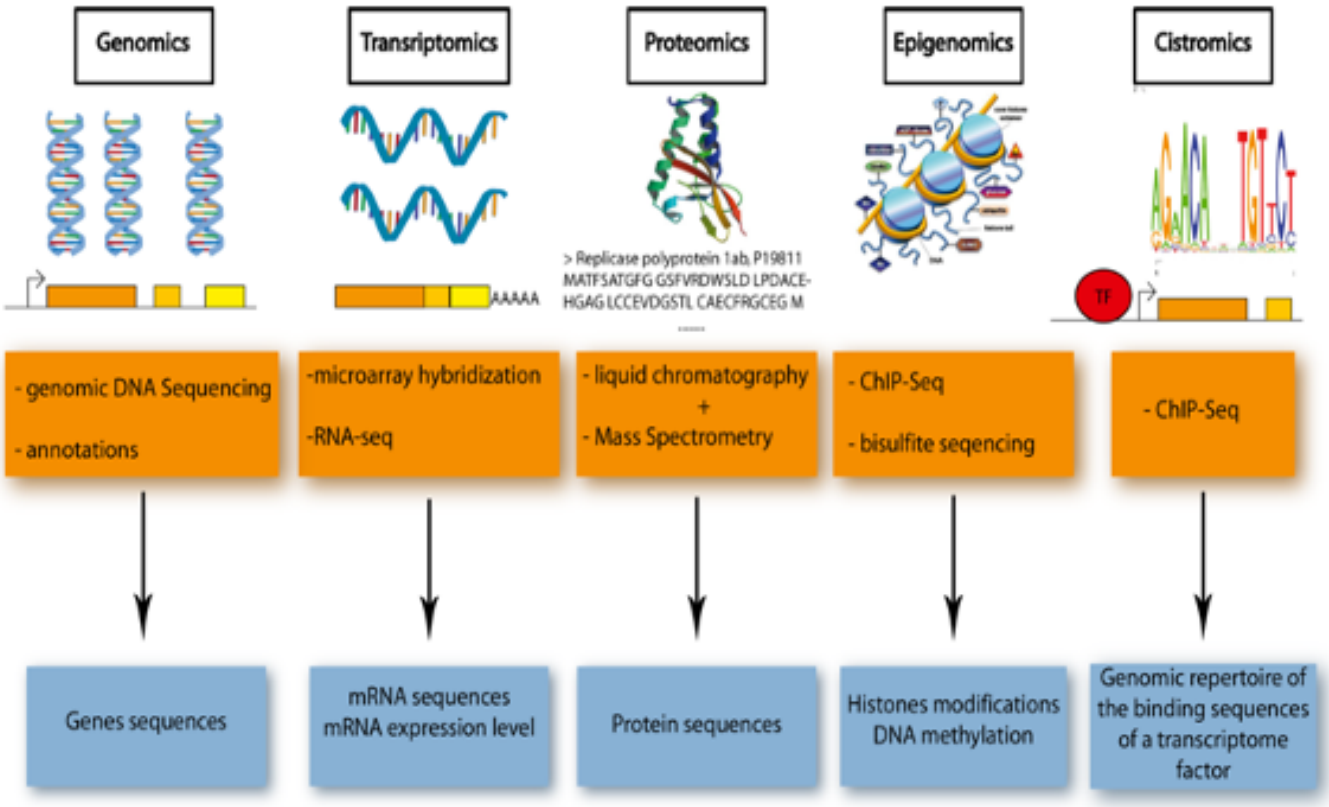


Courtesy Alexander König, Volker Ellenrieder UMG 2019

The Holy Grail of Precision Medicine

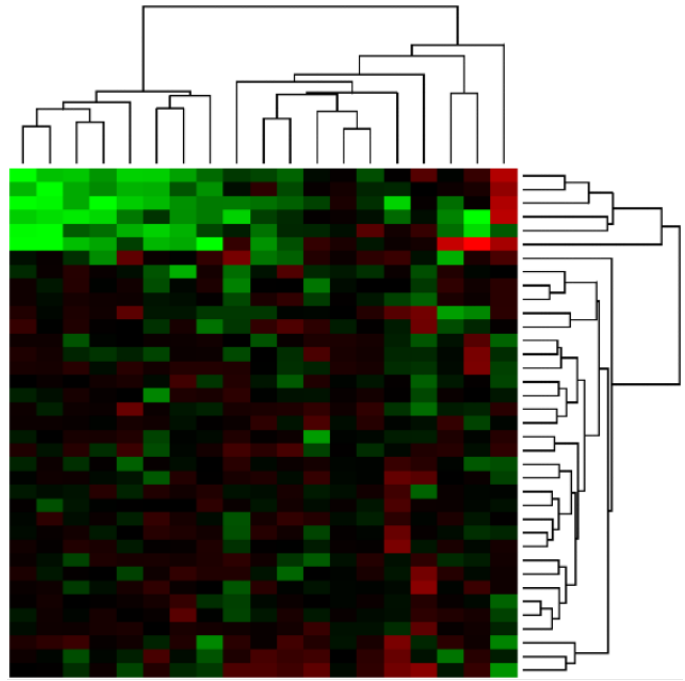
- **Goal:** Understand molecular mechanisms driving complex disease.
  - ⇝ Might pave the way for novel, causally effective treatment strategies.
- **How to:**
  - Mine omics data for candidate disease mechanisms.
  - Validate the predicted mechanisms in silico to assess initial plausibility.
  - Validate the most promising candidate mechanisms in pre-clinical studies.

Reminder: Different Types of Omics Data



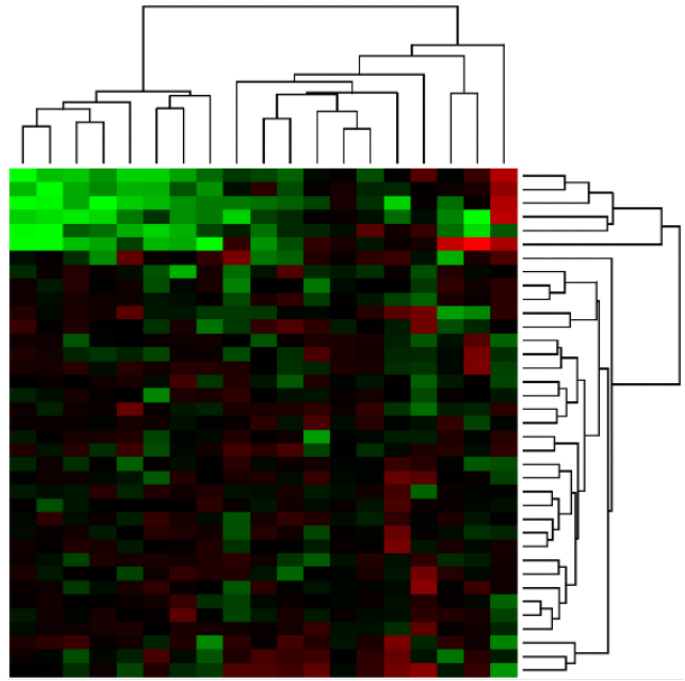
Classical Statistical Data Mining on Omics Data?

- Large number of variables (e.g. genes).
- Small number of samples.
  - $\rightsquigarrow$ Very bad sample-to-feature ratio.
    - Noise.
    - **Solution: gather billion of data points?**

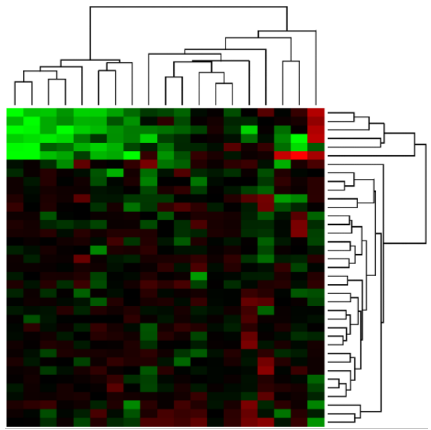


Classical Statistical Data Mining on Omics Data?

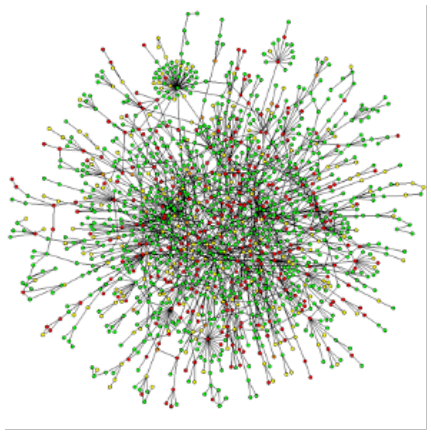
- Large number of variables (e.g. genes).
- Small number of samples.
  - ⇝ Very bad sample-to-feature ratio.
    - Noise.
    - **Solution: gather billion of data points?**
- **Standard analyses ignore effects of interactions.**



Integrating Omics and Network Information



+



- Mitigates shortcomings of purely statistical analyses.
- Find novel potential mechanisms, biomarkers, and drug targets.

⇝ Paves the way for personalized medicine.

Disease Module Mining

Generic Problem Formulation

Given omics data for a disease of interest and a biological network, design a method to compute one or several subnetworks (disease modules) within the biological network that constitute promising candidate disease mechanisms.

Design Decisions to be Made

- Which exact input should our method accept?
- Which algorithmic model should our method employ?

Which Molecular Networks?

PPI Networks

- Can pinpoint to interacting proteins driving a disease.
- \+ Easy to obtain from public databases.
- \+ Experimentally confirmed interactions.
- – Subject to study bias.
- – Often unspecific.

GRNs

- Can pinpoint to gene regulatory cascades that break down in a disease.
- – Only very incomplete GRNs are experimentally confirmed.
- – Typically inferred from gene expression data.
- – Very noisy.
- \+ De novo inferred and therefore not subject to study bias.
- – More targeted, tissue-specific information.

Projecting the Omics Data onto the Networks: Seed Genes Obtained from Gene Expression Data or Genome-Wide Association Studies

- Compute gene-level *P*-values from gene expression or GWAS data.
- Set significance cutoff and keep significant genes as seeds (binary input).
- For many diseases, similarly computed lists of seed genes can be obtained from public databases.

\+ Very simply and user-friendly input format.

– Possible information loss due to binarization.

– Arbitrariness in selecting the cutoff.

Input Used by Existing Tools

Gene Expression Matrix

- COSINE
- GXNA
- GrandForest

Indicator Matrix

- KeyPathwayMiner

Gene Scores

- HotNet2
- Hierarchical HotNet
- NetCore

Seed Genes

- ClustEx2
- **DIAMOnD**
- MuST
- DOMINO
- **ROBUST**

DIAMOnD

- One of the most widely used DMMM (based on number of citations).
- Expects undirected networks (e. g., PPI networks) and seed genes as input.

RESEARCH ARTICLE

A DIseAse MOdule Detection (DIAMOnD) Algorithm Derived from a Systematic Analysis of Connectivity Patterns of Disease Proteins in the Human Interactome

Susan Dina Ghiassian[1,2☯], Jörg Menche[1,2,3☯], Albert-László Barabási[1,2,3,4*]

DIAMOnD

Task

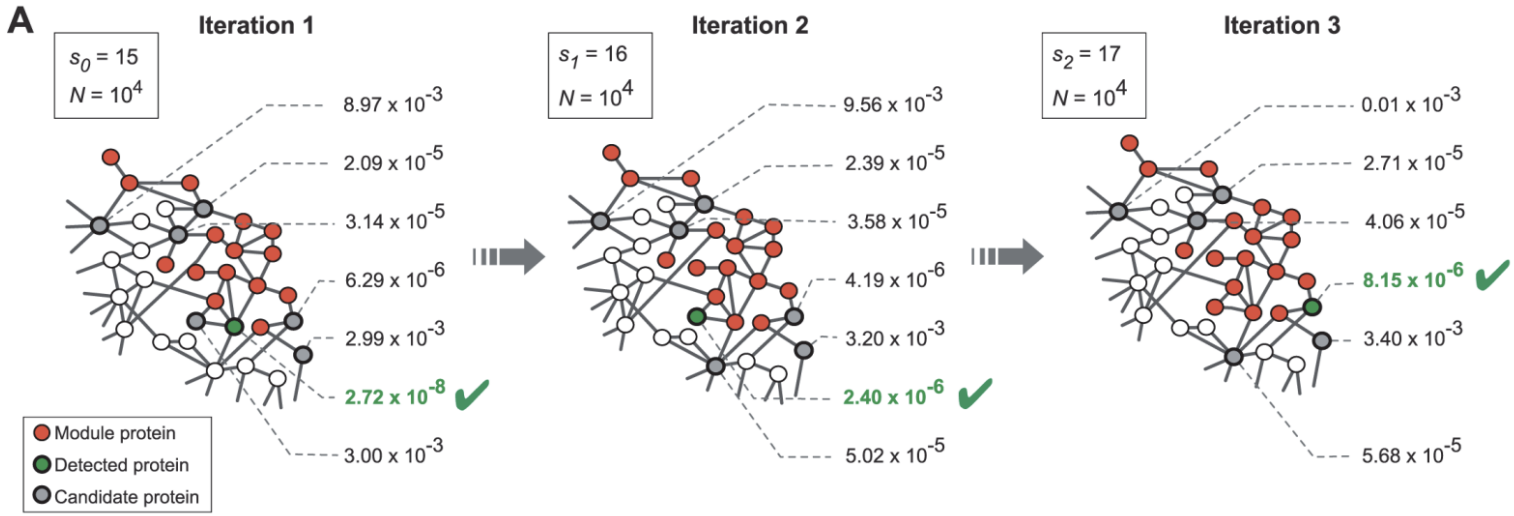
Given biological network $G = (V, E)$ and set of disease-associated seed genes $S \subset V$, find module $G[M]$ constituting a promising candidate disease mechanism, where $M \subset V$ is a node set with $M \supset S$.

Approach

1. **Assumption:** Treat seed sets S as proxies for modules M that should be predicted.
2. **Data integration:** Collect seed sets for many complex diseases.
3. **Data analysis:** Find network property characterizing these seed sets.
4. **Algorithm design:** Use this property to extend the seed sets to disease modules containing them.

The DIAMOnD Algorithm

1. Initialize module M with seed set.
2. Compute connectivity significance for all genes adjacent to a node in M but not contained in it.
3. Put node with smallest P-value into M.
4. Iterate until desired module size has been reached.



Connectivity Significance

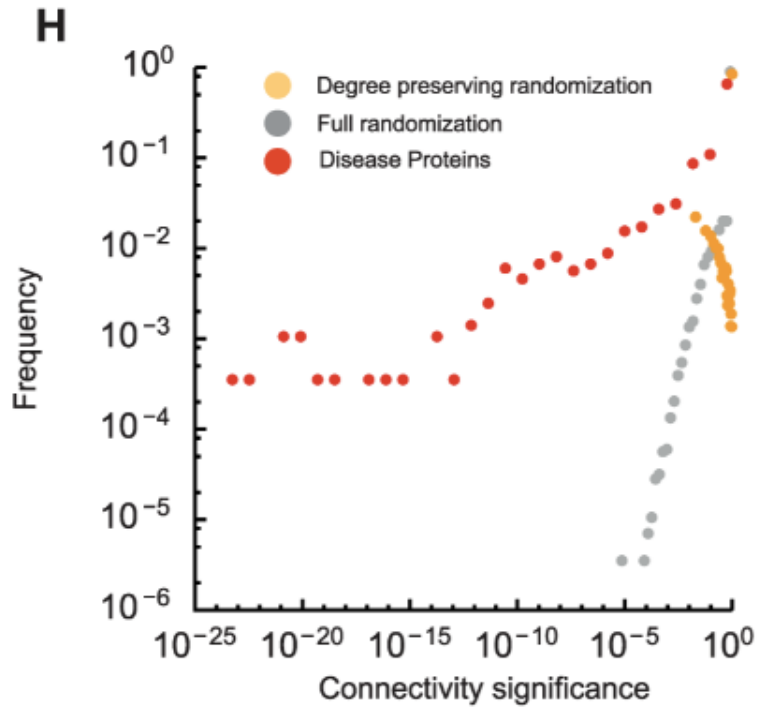
- Consider network with N genes (nodes) and $s_0 < N$ disease-associated seed genes.
- Assume that seed genes are randomly scattered across the network.
- It checks: "Is this protein more connected to the known disease genes than we'd expect just by chance, given its degree?"

Connectivity Significance of Disease-Associated Genes (I)

Comparison Against Random Genes and Randomized Networks

- Compute connectivity significance *P*-values for disease genes.
- Compare against randomly sampled genes.
- Compare agains disease genes in networks randomized with configuration model (degree-preserving).

⇝ Connectivity *P*-values are much smaller for disease genes than for random genes or randomized networks.



BIONETS

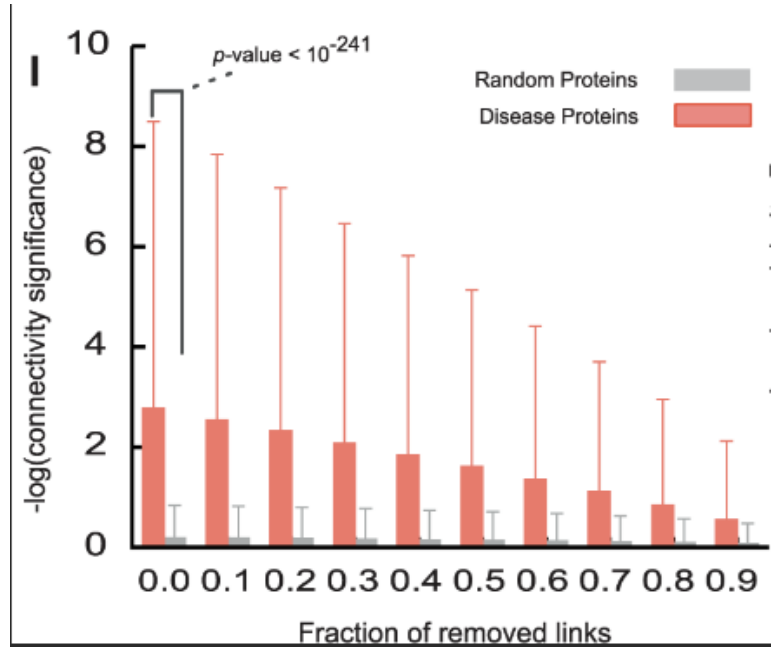
FAU

Connectivity Significance of Disease-Associated Genes (II)

Comparison Against Random Gene Sets with Random Removal of Edges

- Connectivity significance of disease genes drops with increasing number of removed edges.
- Connectivity significance of random genes remains largely unchanged.

⇝ Connectivity significance seams to be the right measure to characterize disease module.



Properties of the DIAMOnD Algorithm

- Genes contained in predicted modules have high connectivity significance by design.
- Implicitly assumes that known seeds constitute the core of the module (modules grow in a BFS-like way).
- All initial seeds always end up in the predicted module.
- Desired module size is a parameter that has to be provided by the user.

ROBUST

- DMMM from our lab.
- Also expects undirected networks (e. g., PPI networks) and seed genes as input.
- Designed to overcome robustness deficit of previous DMMMs.

Bioinformatics

Issues Advance articles Submit ▾ Purchase Alerts About ▾ All Bioinformati

Article Contents

Abstract

Author notes

Supplementary data

ACCEPTED MANUSCRIPT

Robust disease module mining via enumeration of diverse prize-collecting Steiner trees

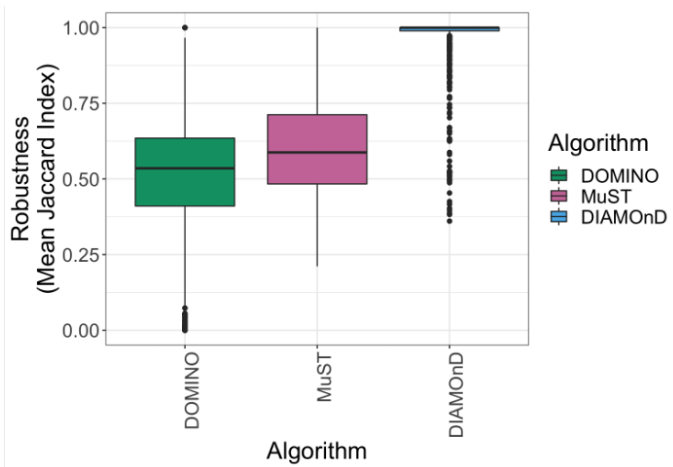
Judith Bernett, Dominik Krupke, Sepideh Sadegh, Jan Baumbach, Sándor P Fekete, Tim Kacprowski, Markus List, David B Blumenthal ✉ Author Notes

Bioinformatics, btab876, https://doi.org/10.1093/bioinformatics/btab876

Published: 04 January 2022 **Article history** ▾

Why Another Method?

- **Robustness of DMMMs:** Mean Jaccard index ($|M \cap M'|/|M \cup M'|$) of modules computed by DMMM when run multiple times on equivalent input (e. g., permuted storage order of network).
- DMMMs lack robustness and are subject to random bias.
- Robustness is important for trustworthiness of DMMMs.
- Biomedical scientists often hesitant to invest time and money in wet-lab validation of non-robust disease modules.
- **Question:** How can the robustness of DMMMs be improved?



Naïve Approach

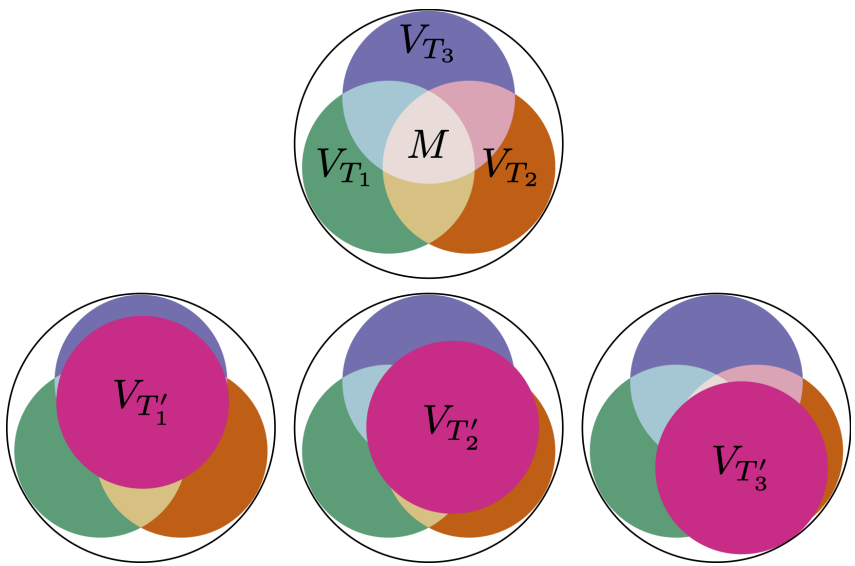
1. Repeat n times:
   - Shuffle input.
   - Run preferred DMMM to obtain disease module M_i.
2. Return subgraph $G[M]$ induced by nodes contained in at least a fraction $\tau \in (0, 1]$ of computed modules ($M := \{v \in V \mid |\{i \mid v \in M_i\}| \geq \tau \cdot n\}$).

Problems

- Increases runtime by a factor of n.
- The modules M_i might not be sufficiently diverse to ensure robustness.

Abstract Approach

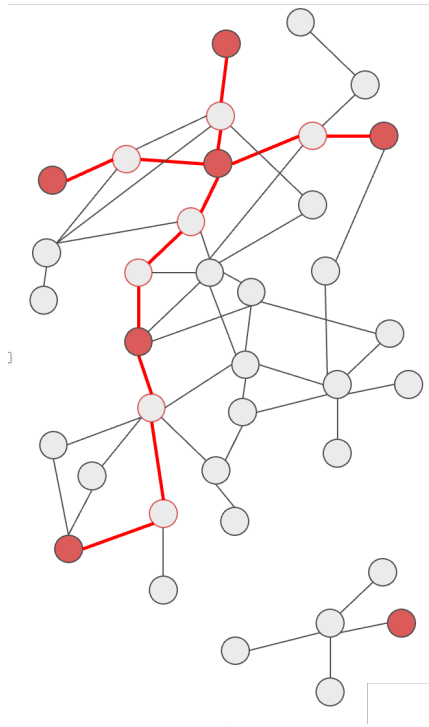
1. Model disease module mining problem as suitable mathematical optimization problem.
2. Design an algorithm to enumerate a diverse set of near-optimal solutions.
3. Return subgraph induced by nodes contained in many of the diverse solutions (as in naïve approach).



The Model Employed by ROBUST

Minimum-Weight Steiner Trees (MWSTs)

- Given (edge-weighted) graph $G = (V, E, w)$ and seeds $S \subseteq V$, compute tree $T = (V_T, E_T)$ with $S \subseteq V_T$ minimizing $\sum_{e \in E_T} w(e)$.
- *NP*-hard but efficient approximation algorithms exist.
- Unit edge weights in our case: $w \equiv 1$.



Prize-Collecting Steiner Trees

- Given graph $G = (V, E, w, p)$ with non-negative edge weights and node prizes, compute tree $T = (V_T, E_T)$ minimizing the following objective:

$$\min \underbrace{\sum_{e \in E_T} w(e)}_{\text{minimize}} + \underbrace{\sum_{v \in V \setminus V_T} p(v)}_{\underbrace{\sum_{v \in V} p(v)}_{\text{constant}} - \underbrace{\sum_{v \in V_T} p(v)}_{\text{maximize}}}$$

- Again *NP*-hard, but (very) efficient approximation algorithms exist.
- **Prizes for seeds:** High but not infinite to encourage inclusion of seeds.
- **Prizes for non-seeds:** Low but not zero, used to ensure diversity.

Enumerating Diverse Prize-Collecting Steiner Trees

- **Line 1:** Initialize prizes for seeds based on diameter (longest shortest path) of graph (and maximum weight).
- **Line 2:** Initialize prizes for non-seeds based on hyper-parameter $\alpha \in (0, 1]$ (and minimum weight).
- **Line 4:** Enumerate up to n prize-collecting Steiner tree.
- **Lines 5 – 7:** Compute next Steiner tree. Break if already seen before.
- **Line 8:** Decrease prizes for non-seeds used in previous Steiner tree by factor $\beta \in [0, 1)$ to ensure diversity.

Algorithm 1: ROBUST

Input: Graph $G = (V, E)$, seeds $S \subseteq V$, parameters $n \in \mathbb{N}$, $\alpha \in (0, 1]$, $\beta \in [0, 1)$, $\tau \in (0, 1]$.
Output: Robust disease module for seeds S.

1 $\mathcal{T} \leftarrow$ `enumerate_diverse`(G, S, n, α, β);
2 $M \leftarrow \{v \in V \mid |\{V_T \in \mathcal{T} \mid v \in V_T\}| \geq \tau \cdot |\mathcal{T}|\}$;
3 **return** $G[M]$;

Algorithm 2: `enumerate_diverse`

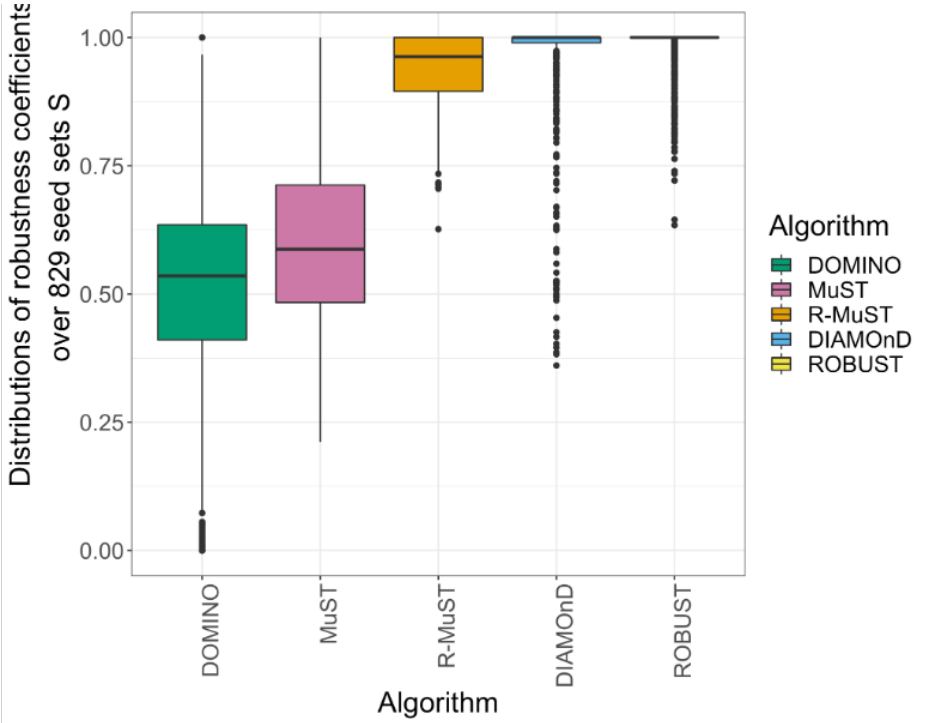
Input: Graph $G = (V, E)$, seeds $S \subseteq V$, edge weights $w : E \to \mathbb{R}_{\geq 0}$, parameters $n \in \mathbb{N}$, $\alpha \in (0, 1]$, $\beta \in [0, 1)$.
Output: Set $\mathcal{T}$ of diverse PCST node sets.

1 **for** $v \in S$ **do** $p(v) \leftarrow 2 \cdot \mathrm{diam}(G) \cdot \max_{e \in E} w(e)$;
2 **for** $v \in V \setminus S$ **do** $p(v) \leftarrow \alpha \cdot \min_{e \in E} w(e)$;
3 $\mathcal{T} \leftarrow \emptyset$;
4 **while** $|\mathcal{T}| < n$ **do**
5 $\quad (V_T, E_T) \leftarrow$ `pcst_apx`(G, w, p);
6 $\quad$ **if** $V_T \in \mathcal{T}$ **then break** ;
7 $\quad \mathcal{T} \leftarrow \mathcal{T} \cup \{V_T\}$;
8 $\quad$ **for** $v \in V_T \setminus S$ **do** $p(v) \leftarrow \beta \cdot p(v)$;
9 **return** $\mathcal{T}$

Effect of the Hyper-Parameters

- $n \in \mathbb{N}$: Coverage vs. runtime. Increasing n increases the runtime but also the fraction of the space of all near-optimal generalized Steiner trees covered by ROBUST. Should be set to high value if affordable.
- $\tau \in (0, 1]$: Explorativeness vs. robustness. Increasing τ increases the robustness but decreases the explorativeness. Should be set to smallest possible value such that robustness is still acceptable.
- $\alpha \in (0, 1]$: Increasing α increases the allowed diversion from cheapest Steiner tree. Should be set to smallest possible value such that robustness is still acceptable.
- $\beta \in [0, 1)$: Controls the decrease of prizes for non-seeds and thereby affects diversity. Must be chosen via hyper-parameter tuning.

Robustness in Comparison to Competitors



Mann-Whitney *U* Test

Alternative hypothesis: DMMM 1 yields larger robustness coefficients than DMMM 2.

DMMM 1	DMMM 2	*P*-Value
ROBUST	DOMINO	1.668×10^{-278}
ROBUST	MuST	3.847×10^{-226}
ROBUST	R-MuST	4.460×10^{-68}
ROBUST	DIAMOnD	6.796×10^{-6}

Drug Prioritization via Personalized PageRank (PPR)

Augment G with drugs to enable drug scoring via PPR.

- **Proteins (PPI):** $G = (V, E)$, undirected, possibly weighted. Adjacency $A \in \mathbb{R}^{|V|\times|V|}$ with $A_{ij} = A_{ji} \geq 0$.
- **Drugs:** set D. Each drug $d \in D$ targets a subset $T(d) \subseteq V$ (drug–target relations).
- **Edges involving drugs:**

$$E_{DT} = \{(d, t) : d \in D,\ t \in T(d)\},$$

- **Augmented undirected graph:**

$$G^+ = (V^+, E^+), \quad V^+ = V \cup D, \quad E^+ = E \cup E_{DT} \cup E_{DD}.$$

Let $A^+ \in \mathbb{R}^{|V^+|\times|V^+|}$ be the symmetric adjacency of G^+ (weights allowed).

- **Walk matrix (column-stochastic):**

$$d_j^+ = \sum_i A_{ij}^+, \quad P_{ij}^+ = \frac{A_{ij}^+}{d_j^+} \quad \text{(so each column of } P^+ \text{ sums to 1).}$$

Drug Prioritization via Personalized PageRank (PPR)

- **Personalization on disease seeds (genes only):**

$$p_i = \begin{cases} \frac{1}{|S|}, & i \in S \\ 0, & i \notin S \end{cases} \qquad \left(p \in \mathbb{R}^{|V^+|},\ \sum_i p_i = 1\right).$$

Personalized PageRank (PPR) on G^+:

$$\boxed{r \;=\; \alpha P^+ r \;+\; (1-\alpha)\, p} \qquad \Longleftrightarrow \qquad r \;=\; (1-\alpha)\,(I-\alpha P^+)^{-1} p,$$

with damping $\alpha \in (0,1)$, $r \in \mathbb{R}^{|V^+|}$, $r_i \geq 0$, $\sum_i r_i = 1$.

Interpretation (undirected G^+): r is a *proximity/affinity* from the seed set S to all nodes in $V^+ = V \cup D$ via many short random-walk paths (longer paths exponentially down-weighted by α).

PPR on the Augmented Graph & Drug Scoring

Drug prioritization: read out scores on drug nodes.

$$\boxed{score(d) \;:=\; r(d), \qquad d \in D}$$

Higher $score(d)$ means drug d lies closer to the disease seeds S through protein targets and network connectivity.

Acknowledgements

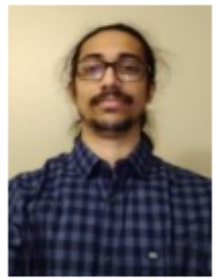
David B. Blumenthal

Head of Lab

Nürnberger Str. 74

91052 Erlangen

Germany



Suryadipto Sarkar, MSc

PhD Candidate

Nürnberger Str. 74

91052 Erlangen

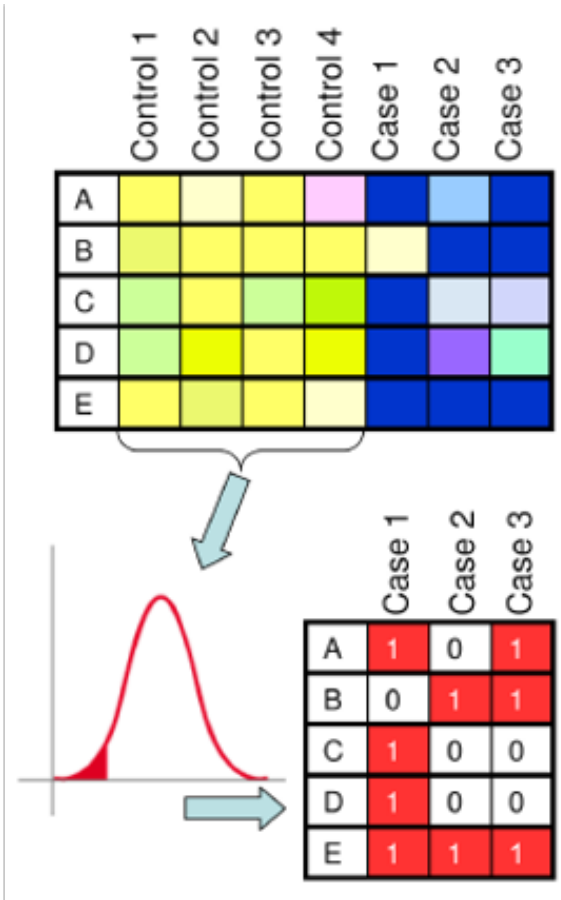
Projecting the Omics Data onto the Networks (I): Directly Using the Gene Expression Matrix

- Let $\mathcal{G}$ be set of genes and $\mathcal{P}$ be set of patients.
- Gene expression matrix is of form $\mathsf{X} \in \mathbb{R}^{\mathcal{G} \times \mathcal{P}}$.
- In our molecular networks, nodes correspond (or can be mapped) to genes.

$\rightsquigarrow$ Can use row $\mathsf{X}_{g,\bullet}$ as $|\mathcal{P}|$-dimensional node attribute of node (gene) $g \in \mathcal{G}$.

- If phenotype case/control information is available, we additionally store a global indicator vector $\mathsf{b} \in \{0,1\}^{\mathcal{P}}$, where $b_p = 1$ if and only if $p \in \mathcal{P}$ is a case patient.

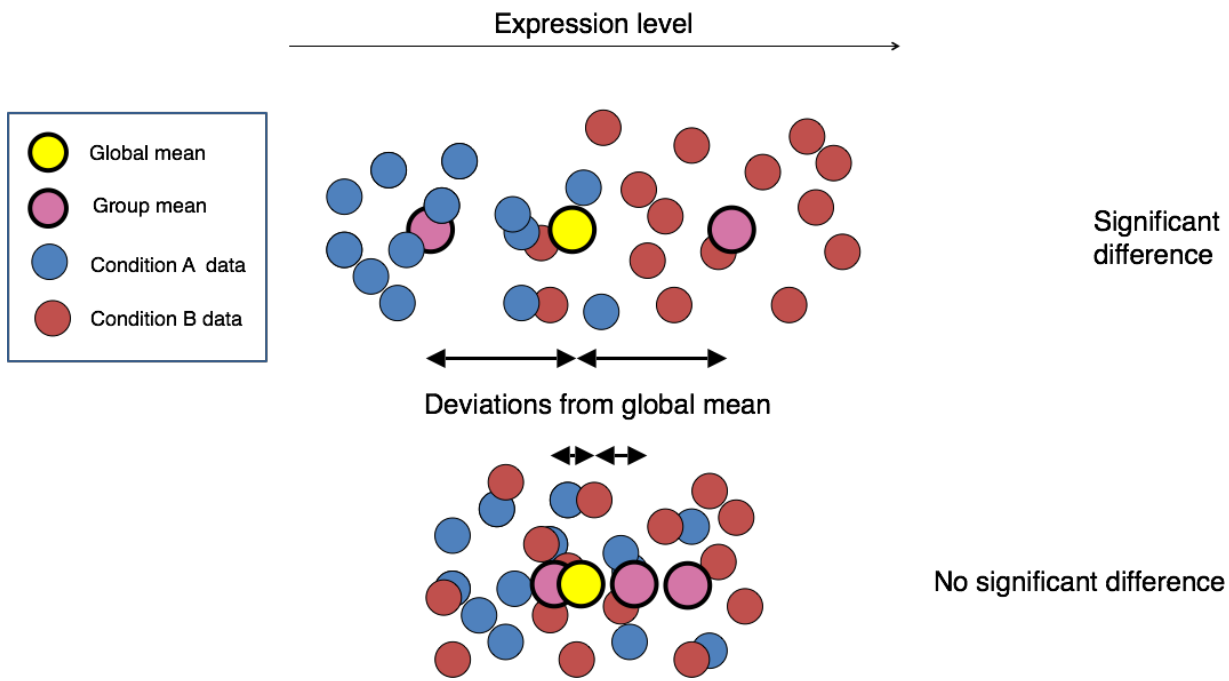
Projecting the Omics Data onto the Networks (II): Using an Indicator Matrix Derived from the Gene Expression Matrix



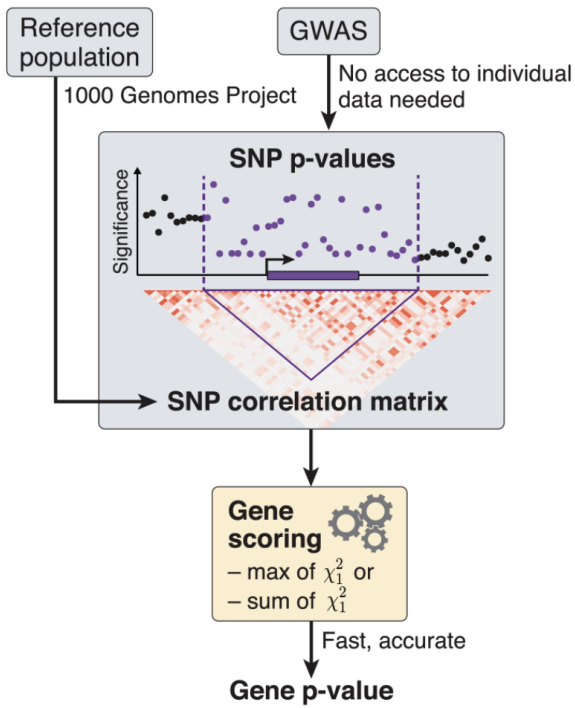
- For each gene, fit background gene expression distribution based on the control samples only.
- For each gene and each case patient, compute *P*-value for diversion from background.
- Select threshold for significance cutoff.

Projecting the Omics Data onto the Networks (III): Gene Scores Obtained from Gene Expression Data

- For each gene, compute P-value of differential expression in case samples.
- Use $-\log P$-value as gene score.



Projecting the Omics Data onto the Networks (IV): Gene Scores Obtained from Genome-Wide Association Studies (GWAS, Genomics Data)



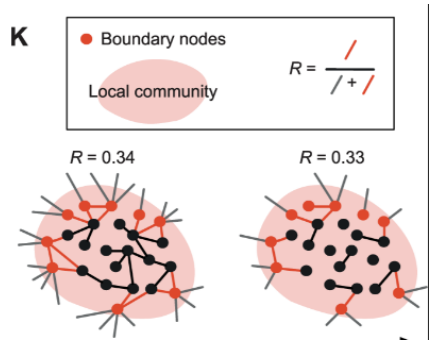
- GWAS yield *P*-values for individual genetic variants.
- Aggregate *P*-values for variants associated to gene to obtain gene-level *P*-values.
- Use $-\log P$-value as gene score.

One Possible Property: Local Modularity

- For some module M, let $B \subset M$ be boundary of M, i. e., the set of nodes u with at least one neighbor $v \notin M$.
- Let $T := \{uv \in E \mid u \in B \vee v \in B\}$ be set of all edges incident with a boundary node.
- Let $I := \{uv \in E \mid u \in B \vee v \in B \wedge u, v \in M\}$ be set of all edges incident with a boundary node and both endpoints in module M.
- Note that $I \subsetneq T$.
- Then the local modularity R is defined follows:

$$R := \frac{|I|}{|T|}$$

- Note that $0 \leq R < 1$.

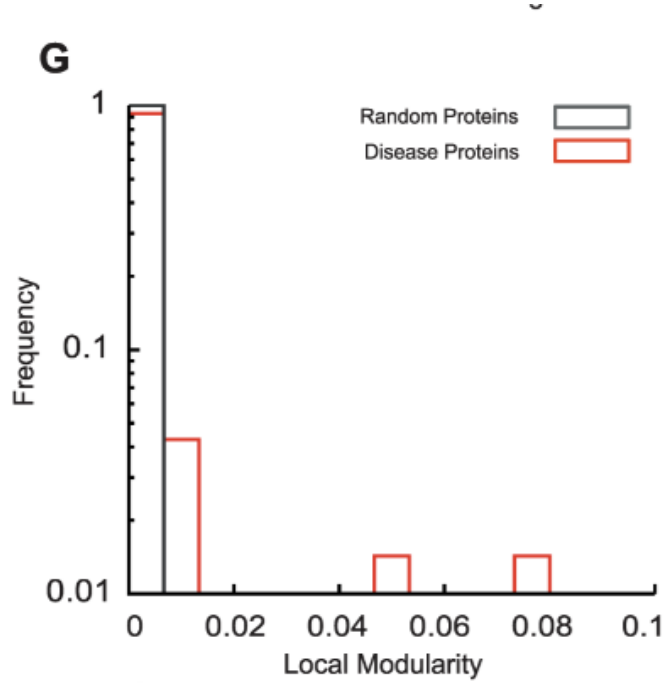


Local Modularity of Sets of Disease-Associated Genes (I)

Comparison Against Random Gene Sets

- Compute local modularity R for sets of disease genes.
- Compare against random gene sets.

$\rightsquigarrow$ R is significantly larger for disease genes but still pretty close to 0.

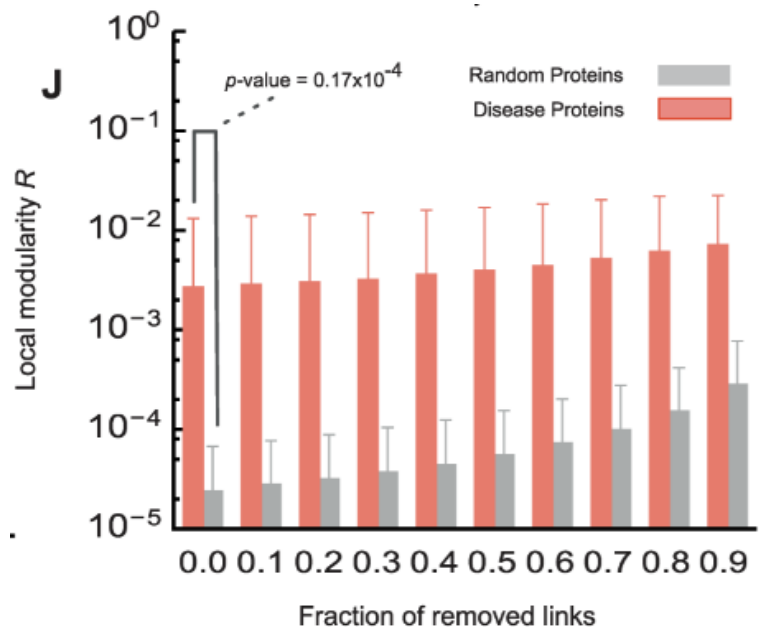


Local Modularity of Sets of Disease-Associated Genes (II)

Comparison Against Random Gene Sets with Random Removal of Edges

- Randomly remove edges from the network.
- If R was a good measure to characterize disease modules, it should drop with increasing number of removed edges.
- This is not the case.

$\rightsquigarrow$ R might not be the best measure to characterize disease module.



Connectivity Significance

- Consider network with N genes (nodes) and $s_0 < N$ disease-associated seed genes.
- Assume that seed genes are randomly scattered across the network.
- What is the probability $P_{N,s_0}(k, k_s)$ that a gene with with k neighbors has exactly $k_i \leq k$ neighbors among the s_0 seed genes?
- Given by hypergeometric distribution:

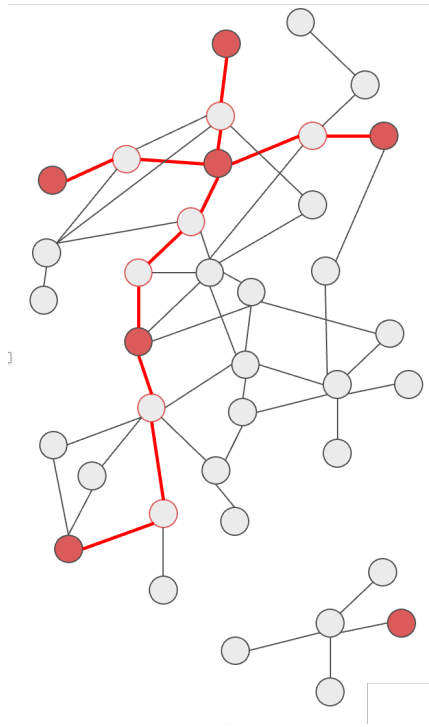
$$P_{N,s_0}(k, k_i) = \frac{\binom{s_0}{k_i}\binom{N-s_0}{k-k_i}}{\binom{N}{k}}$$

$\rightsquigarrow$ **Connectivity P-value:** Probability under null-model that node with k neighbors has at least k_s neighbors among the seeds:

$$P_{N,s_0}(k, k_i \geq k_s) = \sum_{k_i=k_s}^{k} P_{N,s_0}(k, k_i)$$

From Generalized MWSTs to Prize-Collecting Steiner Trees

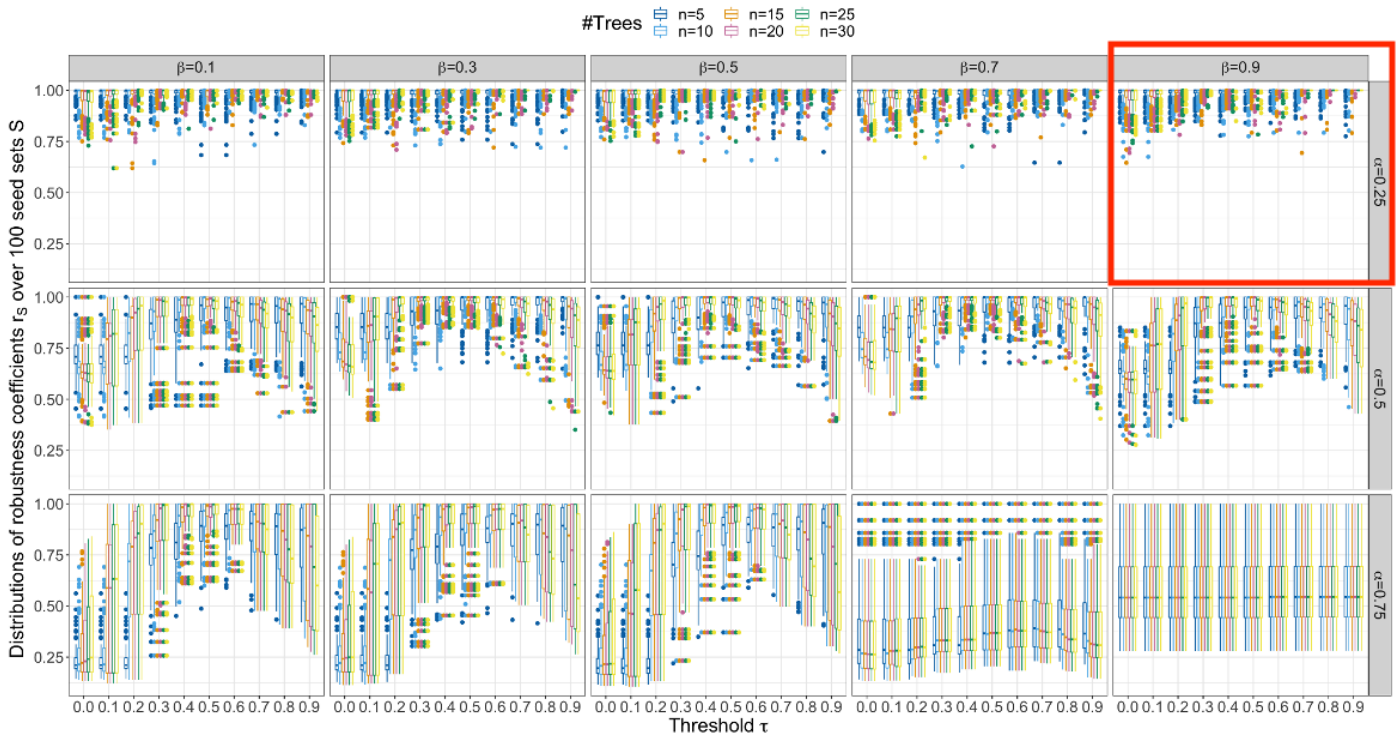
- Our seeds are potentially noisy (e. g., false positives in GWAS).
- We do not enforce $S \subseteq V_T$ but only encourage the inclusion of seed.
- Seeds that are very far away from the other seeds will not be connected in the module.
- To achieve this, we use **prize-collecting** Steiner trees instead of MWSTs.



Hyper-Parameter Tuning for Robustness (I)

- Increasing α decreased robustness, increasing β increased it.

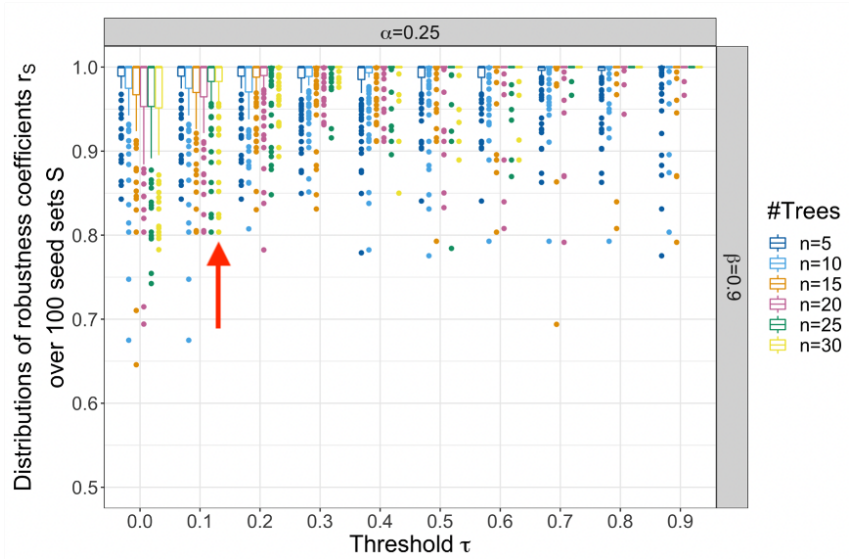
$\rightsquigarrow$ Focus on $\alpha = 0.25$ and $\beta = 0.9$.



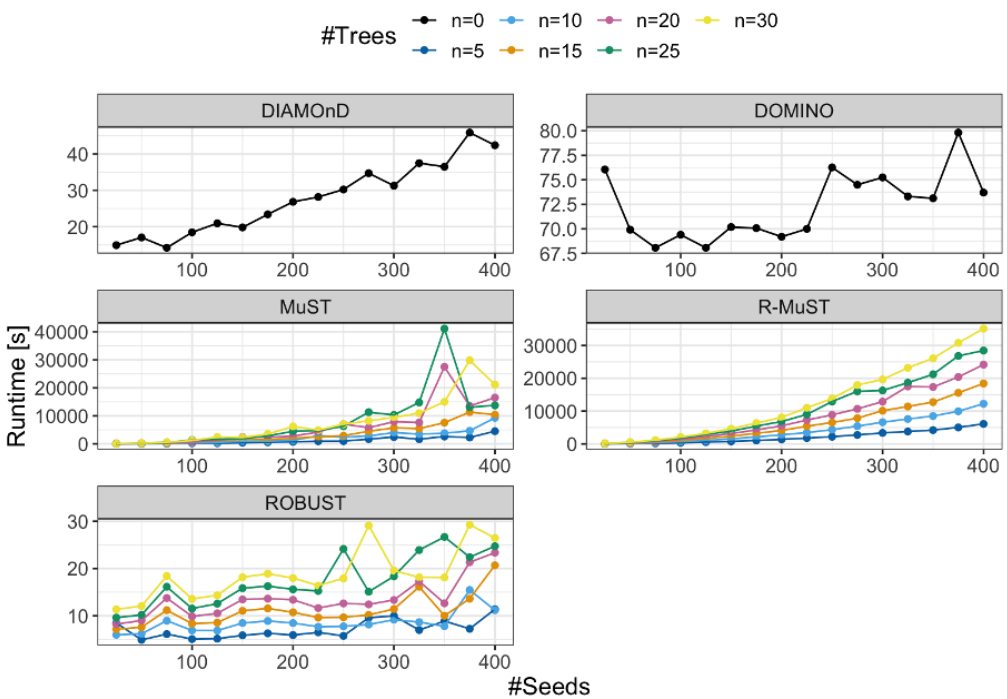
Hyper-Parameter Tuning for Robustness (II)

- Very good robustness already for $\tau = 0.1$ and $n = 30$.
- Runtime still acceptable for $n = 30$ trees.

$\rightsquigarrow$ Use hyper-parameter setup $(\alpha, \beta, n, \tau) = (0.25, 0.9, 30, 0.1)$ for comparison against competitors.

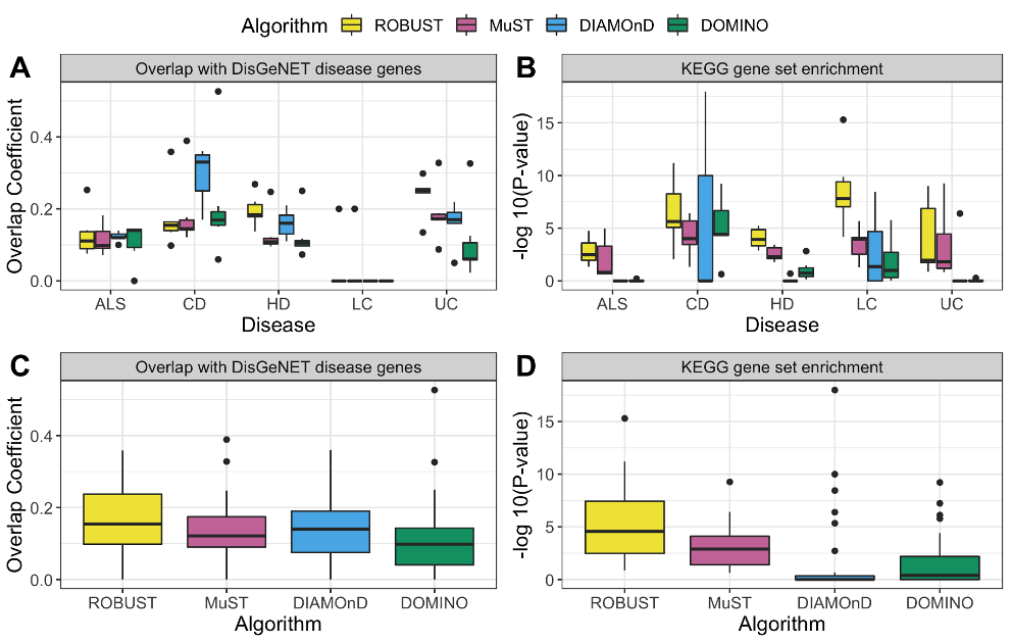


Runtime in Comparison to Competitors



- MuST and R-MuST around 2 orders of magnitude slower.
- Runtime increases sub-linearly for ROBUST and DOMINO, linearly for DIAMOnD.
- Number of trees affect ROBUST's runtime very moderately.

Functional Relevance in Comparison to Competitors



- Seeds derived from gene expression data for five complex diseases.
- Tests run on five different PPI networks (from databases).
- Functional relevance measure via overlap w. r. t. disease genes from databases (KEGG, DisGeNET).