Supplementary information

Multi-Objective Gene Selection

Problem Definition

The multi-objective gene selection problem consists of finding optimal subsets of genes from a given dataset with respect to multiple objectives, such as sample coverage, alteration overlap and subset size. Found subsets represent optimal trade-offs between the chosen objectives. The multi-objective gene selection problem is based on binary alteration data, such as mutation data, binarized expression data or binarized methylation data. For a given set of samples \( \Theta = \{ \theta_1, \ldots, \theta_n \} \) and a given set of genes \( G = \{ g_1, \ldots, g_m \} \), the set of samples with an alteration in \( g \in G \) is denoted as \( \Gamma(g) \subseteq \Theta \). A sample \( \theta \in \Theta \) is said to be covered by a gene \( g \in G \) if and only if the sample \( \theta \) has an alteration in the gene \( g \), written as \( \theta \in \Gamma(g) \).

A sample \( \theta \in \Theta \) is covered by a set of genes \( G \subseteq G \) if and only if the sample is covered by one of the genes in \( G \), i.e. \( \theta \in \bigcup_{g \in G} \Gamma(g) = \Gamma(G) \). The coverage \( \gamma(g) \) of a gene \( g \in G \) is the number of samples with an alteration in \( g \), i.e. \( \gamma(g) = |\Gamma(g)| \). The coverage \( \gamma(G) \) of a set of genes \( G \subseteq G \) is the number of different samples that have alterations in at least one of the genes in \( G \):

\[
\gamma(G) = \left| \bigcup_{g \in G} \Gamma(g) \right|
\]

and the overlap of \( G \) is defined as

\[
\omega(G) = \left( \sum_{g \in G} \gamma(g) \right) - \gamma(G) .
\]

This overlap definition accounts for the number of additional genes in \( G \) that cover the same sample in contrast to counting the number of samples covered by more than one gene from \( G \).

The gene selection task consists in finding a subset of genes \( G \subseteq G \) such that the number of covered samples \( \gamma(G) \) is maximized and either the overlap \( \omega(G) \) or the number of genes in \( G \) is minimized. There is no obvious choice for a weighting between the two optimization objectives. A setup using equal weights on unnormalized objectives has been used previously [1]. Hence, the described research questions are formulated as a multi-objective gene selection task that consists of finding a Pareto-optimal set \( S^* \subseteq S \) of gene subsets within the set of all gene subsets \( S = \{ G \subseteq G \} \). The solutions, i.e. gene subsets, in the Pareto-optimal set \( S^* \) are the optimal trade-offs between the two optimization objectives [2]. Pareto-optimality is based on the dominance relation \( \prec \), i.e. a solution \( G \) dominates another solution \( G' \), denoted as \( G \prec G' \), if and only if \( G \) is at least as good as \( G' \) in all objectives and strictly better in at least one. Based on this, the Pareto-optimal set is defined as the set of solutions that are not dominated by any other solution:

\[
S^* = \left\{ G' \in S \mid \exists G \in S : G \prec G' \right\} .
\]

Figure 1 shows a Pareto-optimal set for a multi-objective optimization task maximizing coverage and minimizing overlap. The solutions dominated by the Pareto-optimal solution \( G' \) are marked.

Multi-Objective Evolutionary Algorithm

We developed an evolutionary algorithm for the multi-objective gene selection task based on the Non-dominated Sorting Genetic Algorithm II [2, NSGA-II] implemented by jMetal library v.5.3 [3]. This is a population-based metaheuristic which adapts concepts of the theory of evolution [4]. A set of solutions, called population, is evolved iteratively by applying recombination and mutation operators to the solutions (see Figure 2). The algorithm uses the following binary solution encoding. Genes are sorted by sample coverage in decreasing order. The \( i \)-th bit in a solution \( s \) encodes whether the \( i \)-th gene \( g \) is part of the gene set \( G \) encoded by this solution. Our NSGA-II for Multi-Objective Gene Selection is outlined in Algorithm 1. The algorithm starts with an initial population of \( \mu \in N \) randomly created solutions with a probability of \( \pi_{ext} \in (0, 1) \) to set a bit to 1 (otherwise 0). Tournament selection with tournament size \( \tau \in N \) and the crowded-distance comparator are used to select solutions from the parent population for recombination and mutation. The crowded distance \( d(s_i) \) of a solution \( s_i \) is calculated based on the permutations \( \pi_j \) resulting from sorting the solutions by the \( j \)-th objective \( f_j \) as follows:

\[
d(s_i) = \sum_{j=1}^{k} \frac{f_j(s_{\pi_j(i+1)}) - f_j(s_{\pi_j(i-1)})}{f_j^{\max} - f_j^{\min}}
\]

and \( d(s_{\pi_j(i)}) = \infty \), \( d(s_{\pi_j(\mu)}) = \infty \) for the extrema per objective \( f_j \). With probability \( p_{crossover} \in [0, 1] \) a pair of selected solutions is modified by a single point crossover. The resulting solutions are
Objective Alteration Plots via Algorithmic Sorting

AVAtar facilitates the creation of objective and reproducible alteration plots by offering algorithmic sorting of genes and samples. The task of finding a gene order based on the additionally covered samples is formulated as a minimal Set Cover problem. A modified greedy algorithm [9] for Set Covering is applied (see Algorithm 2). Starting with an initial solution, the greedy algorithm incrementally adds the gene covering the most uncovered samples to its current partial solution, i.e. the gene \( g \) that maximizes the objective function \( f_G(g) = \gamma(G \cup \{g\}) \). For a gene order based on maximal coverage and maximal exclusivity (i.e. minimal overlap), this objective function is set to
\[
\alpha \cdot \gamma(G \cup \{g\}) + (1 - \alpha) \cdot (1 - \gamma(G \cup \{g\})) \rightarrow \max
\]
including a weighting \( \alpha \in [0,1] \) between coverage and exclusivity. To permit the weighting between both aims, the modified objective function uses relative coverage \( \gamma^r \) and relative

Algorithm 2: Greedy Gene Sorting

1. \( G_0 \leftarrow \emptyset \), \( G_1 \leftarrow G \), and \( t \leftarrow 1 \)
2. Empty sorted gene list \( \pi \)
3. while \( G_t \neq \emptyset \) do
4. if \( G_{t-1} = \emptyset \) then // initial selection or restart after reaching maximal coverage?
5. \( \gamma^{\max} \leftarrow \gamma(G_t) \)
6. \( \{g, g_{t+1}\} \leftarrow \arg \max \left\{ f_G(g, g') \mid (g, g') \in G_t^\max \cdot \gamma(g) > \gamma(g') \right\} \)
7. \( \pi_t \leftarrow g \), and \( \pi_{t+1} \leftarrow g_{t+1} \)
8. \( G_t \leftarrow G_{t-1} \cup \{g\} \) and \( G_{t+1} \leftarrow G_t \cup \{g_{t+1}\} \)
9. \( G_{t+1} \leftarrow G_{t+1} \setminus \{g\} \) and \( G_{t+2} \leftarrow G_{t+1} \setminus \{g_{t+1}\} \)
10. \( t \leftarrow t + 2 \)
11. else
12. \( g_e \leftarrow \arg \max \left\{ f_G(G_{t-1}, g) \mid g \in G_t \right\} \)
13. \( \pi_t \leftarrow g_e \)
14. \( G_{t+1} \leftarrow G_t \setminus \{g_e\} \)
15. if \( \gamma(G_t) < \gamma^{\max} \) then
16. \( G_{t+1} \leftarrow G_{t-1} \cup \{g_e\} \)
17. else
18. \( G_t \leftarrow \emptyset \) // maximal coverage reached \( \Rightarrow \) restart
19. \( t \leftarrow t + 1 \)
20. return \( \pi \)
Supplement: Analysis, identification and visualization of subgroups in genomics

Fig. 4. Visualization for genomic data of a subset of 1,540 patients from our AML dataset [6]. Genes are grouped into predefined functional categories. In this subgroup of NPM1 mutated patients, most co-mutations occur in genes associated to the functional groups activated signaling and DNA methylation. NPM1 is fixed to the first position and the remaining genes are sorted according to a trade-off between mutually exclusivity and coverage using $\alpha = 0.4$. The sorted alteration plot highlights the molecular heterogeneity across patients. This visualization with AVAtar shows the molecular heterogeneity within these patients clearly.

\[ \text{overlap } \omega' \text{ of the gene set } G \subseteq \mathcal{G} \text{ compared to the set of all genes } \mathcal{G} \text{ defined as} \]
\[ \gamma'(G) = \frac{\gamma(G)}{\gamma(\mathcal{G})} \quad \text{and} \quad \omega'(G) = \frac{\omega(G)}{\omega(\mathcal{G})} \]  

Similarly, sorting genes according to co-occurring alterations and coverage can be achieved with the following objective function:

\[ f_{\text{co}}(G, g) = \alpha \cdot \gamma'(G \cup g) + (1 - \alpha) \cdot \gamma'(G \cup g) \rightarrow \max . \]  

Finally, samples are sorted lexicographically according to the gene order. Figure 4 shows a visualization for genomic data of a subset of 1,540 patients from an AML dataset [6] using AVAtar’s ability to additionally group genes on predefined functional categories. In this subgroup of NPM1 mutated patients, most co-mutations occur in genes associated to the functional groups activated signaling and DNA methylation. NPM1 is fixed to the first position and the remaining genes are sorted according to a trade-off between mutually exclusivity and coverage using $\alpha = 0.4$. The sorted alteration plot highlights the molecular heterogeneity across patients.

References

[1] Vandin, F., Upfal, E. and Raphael, B. J. De novo discovery of mutated driver pathways in cancer. *Genome Res.*, 2012; *22*:375–385.

[2] Deb, K. Multi-Objective Optimization Using Evolutionary Algorithms. NY, USA: John Wiley & Sons, Inc., 2001.

[3] Nebro, A. J., Durillo, J. J. and Vergne, M. Redesigning the jMetal Multi-Objective Optimization Framework. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO Companion ’15. NY, USA: ACM, 1093–1100.

[4] Reeves, C. R. Handbook of Metaheuristics, chapter Genetic Algorithms. Springer US, 2010; 109–139.

[5] Cormen, T. H., Leiserson, C. E., Rivest, R. L. *et al.* Introduction to Algorithms. The MIT Press, 3rd edition, 2001.

[6] Papaemmanuil, E., Gerstung, M., Bullinger, L. *et al.* Genomic classification and prognosis in acute myeloid leukemia. *N Engl J Med*, 2016; *374*:2209–2221.