Scalable Optimization Algorithms for High-throughput Genomic Data

James Richter Lindsay

University of Connecticut, 2015

The problem of interpreting biological data is often cast into a mathematical optimization framework where a large body of existing computational theory and practical techniques can be leveraged. While this strategy has been particularly successful in the bioinformatics domain, the massive datasets generated by high-throughput genomic technologies are challenging the scalability of even the most advanced mathematical optimization algorithms. Indeed, as the cost per base of of DNA sequencing has dropped precipitously, even outpacing Moore's law, the size of many bioinformatics problems has grown beyond the limit of existing methods, necessitating new algorithms. This effect is felt even more acutely in the burgeoning field of single cell biology where advances in microfluidics has rapidly increased the ability of bench biologists to capture and sequence the genomes and transcriptomes of hundreds of cells per experiment.

This dissertation presents novel computational method for answering three distinct biological questions: genome scaffolding, biomarker selection, and computational deconvolution of gene expression data from heterogeneous samples assisted by single-cell expression data. Each method strives to balance computational efficiency with the biological relevance of computed solutions.

Scalable Optimization Algorithms for High-throughput Genomic Data

James Richter Lindsay
BSc, University of Connecticut, 2007
MS, University of Connecticut, 2009

A Dissertation
Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy
at the
University of Connecticut
2015

Preface

Next-generation sequencing, microarrays and other high-throughput technologies have generated an enormous amount of biological data. Analysis of these large datasets poses significant bioinformatics challenges and requires novel algorithms that are accurate and scalable. Further compounding this challenge is the fact that new high-throughput technologies are often error prone. The following body of work explores the use of scalable optimization algorithms for three distinct bioinformatics problems.

Genome scaffolding. At the core of modern genetics studies is availability of the genome sequence. There has been much work in developing scalable genome assembly algorithms that are both accurate and capable of working with high-throughput sequencing data. However it has been observed that extremely high-coverage, at a higher cost, is necessary to obtain contigs long enough to be biological useful. One key component of modern genome assembly pipelines is a scaffolding step whereby assembled contigs are ordered and oriented relative to each other.

Integer linear programming (ILP) is a powerful combinatorial optimization technique that allows modeling and computing optimal solutions of complex real-world problems. One drawback of this approach is that in the worst-case ILP solvers can take exponential time. This work presents an ILP based solution to the scaffolding problem that is both accurate and scalable. Scalability is achieved through the use of Non-serial Dynamic Programming (NSDP), a technique which exploits the natural sparsity of the problem to compute the optimal solution in stages.

Biomarker selection. Another classic bioinformatics problem made challenging due to high-throughput data is the task of building predictive classification models. There can be thousands to millions of potential features, commonly referred to as biomarkers. Often a small yet maximally informative subset of biomarkers is desired due to cost, performance issues, and the desire for simplicity. This problem is known as feature selection.

Given a biological dataset there exists little a priori justification for choosing a particular feature selection and classification algorithm. Experience or anecdotal evidence is often used by bioinformatics researchers to choose an approach to apply to a dataset. A comprehensive comparative study can explores the efficacy of various feature selection and classification algorithms on diverse genomic datasets.

When there is missing data, and the data is binary with no replicates then the problem becomes challenging. This scenario occurs when the features are curated from literature and in-situ experiments (images). An additional practical challenge is to identify the most informative markers given physical constraints dictated by the technology used to interrogate the chosen markers such as the number of wells available on one qPCR chip. We present an ILP feature selection algorithm that has comparable performance to leading statistical approaches while keeping the ability to add real-world constraints.

Single-cell assisted deconvolution. Whole transcriptome expression profiles captured using high-throughput technologies often come from bulk biological samples consisting of heterogeneous mixtures of cells. However, in many contexts it would be beneficial to infer the expression profiles and concentrations of each constituent cell-type. This problem is known as the gene expression deconvolution problem.

Existing deconvolution approaches often perform poorly when constituent cell-types have highly similar expression profiles, e.g., when attempting to discern between a progenitor cell and its recent progeny. The development of single-cell resolution qPCR has allowed for the accurate survey of a small number of genes at the cellular level. Initial work utilizing a quadratic programming (QP) formulation to exploit single-cell qPCR data for deconvolution of heterogeneous bulk samples is presented.

APPROVAL PAGE
Doctor of Philosophy Dissertation

Scalable Optimization Algorithms for High-throughput Genomic Data

James Richter Lindsay
University of Connecticut, 2015

Professor Ion Mandoiu...
Major Advisor

Professor Yufeng Wu..
Associate Advisor

Professor Sanguthevar Rajasekaran ....
Associate Advisor

Professor Alex Zelikovsky..
Associate Advisor

Professor Craig Nelson...
Associate Advisor

Acknowledgments

I would like thank my advisor professor Ion Mandoiu for his continued support over the course of my education. Professor Mandoiu began working with me as an undergraduate and has guided me through this stage of my life. I would also like to thank my wife Emma Lindsay, and my family, without whom none of this would be possible.

Contents

List of Figures

List of Tables

Chapter 1

Genome Scaffolding

Interest in *de novo* genome assembly has been renewed in the past decade due to rapid advances in high-throughput sequencing (HTS) technologies which generate relatively short reads resulting in highly fragmented assemblies consisting of contigs.[1] Additional long-range linkage information is typically used to orient, order, and link contigs into larger structures referred to as *scaffolds*. Due to library preparation artifacts and erroneous mapping of reads originating from repeats, scaffolding remainings a challenging problem. In this paper, we provide a scalable scaffolding algorithm (SILP2) employing a maximum likelihood model capturing read mapping uncertainty and/or non-uniformity of contig coverage which is solved using integer linear programming. A Non-Serial Dynamic Programming (NSDP) paradigm is applied to render our algorithm useful in the processing of larger mammalian genomes. To compare scaffolding tools, we employ novel quantitative metrics in addition to the extant metrics in the field. We have also expanded the set of experiments to include scaffolding of low-complexity metagenomic samples.

SILP2 achieves better scalability through a more efficient NSDP algorithm than previous release of SILP. The results show that SILP2 compares favorably to previous methods OPERA and MIP in both scalability and accuracy for scaffolding single genomes of up to human size, and significantly outperforms them on scaffolding low-

[1] The results presented in this chapter are based on joint work with H. Salooti, I. Mandoiu and A. Zelikovsky published in [62].

complexity metagenomic samples.

Equipped with NSDP, SILP2 is able to scaffold large mammalian genomes, resulting in the longest and most accurate scaffolds. The ILP formulation for the maximum likelihood model is shown to be flexible enough to handle metagenomic samples.

1.1 Motivation

De novo genome assembly is one of the best studied problems in bioinformatics. Interest in the problem has been renewed in the past decade due to rapid advances in *high-throughput sequencing* (HTS) technologies, which have orders of magnitude higher throughput and lower cost compared to classic Sanger sequencing. Indeed, top-of-the-line instruments from Illumina and Life Technologies are currently able to generate in a single run billions of reads with an aggregate length of hundreds of gigabases, at a cost of mere cents per megabase. However, most HTS technologies generate relatively short reads, significantly increasing the computational difficulty of the assembly problem. Despite much work on improved assembly algorithms for HTS shotgun reads [104, 17, 32, 19, 57, 111], *de novo* assembly remains challenging, often resulting in highly fragmented assemblies, see [4, 25, 61, 76, 80, 82, 96, 93] for recent reviews and benchmarking results. For example, the recent Assemblathon 2 community effort to benchmark *de novo* genome assemblers [4] shows that the performance of evaluated assemblers is highly variable from dataset to dataset and generally degrades with the complexity of the sample.

To increase the utility of such fragmented assemblies, additional long-range linkage information is typically used to orient, order, and link contigs into larger structures referred to as *scaffolds*. Although long-range linkage information can be generated using a variety of technologies, including Sanger sequencing of both ends of cloned DNA fragments of up to hundreds of kilobases, Pacific Biosciences reads of up to tens of kilobases [66], and optical maps [77], the most common type of data used in scaffolding are HTS read pairs generated from DNA fragments with length ranging between hundreds of bases to tens of kilobases.

While HTS read pairs are relatively easy to generate, the linkage information they provide is noisy due to library preparation artifacts and erroneous mapping of reads originating from repeats. The general scaffolding problem is known to be computationally NP-hard when linkage data contains errors [46]. Moreover, the associated contig orientation and contig ordering problems are intractable as well: the orientation problem is equivalent to finding a maximum bipartite subgraph, whereas the ordering problem is similar to the Optimal Linear Arrangement problem, both of which are NP-hard [27]. Due to the intractability of the problem, greedy heuristics have been employed in practical scaffolding methods such as[46, 83]. Scaffolding methods such as SOPRA [21] reduce the size of the problem by iteratively removing inconsistent links and contigs, while MIP [92] heuristically partitions the biconnected components of the scaffolding graph when they are too large to scaffold optimally by mixed integer programming. In SLIQ [89], inequalities are derived from the geometry of contigs to predict the orientation and ordering of adjacent contigs. To find a feasible solution with minimum read pair inconsistency, OPERA [26] provides a novel dynamic programming algorithm.

Algorithms based on explicit statistical models are currently gaining popularity in the area of genome assembly [43], with notable advances in the use of *maximum likelihood* (ML) methods for both contig assembly [75] and assembly evaluation [84]. In this paper we introduce a highly scalable algorithm based on likelihood maximization for the scaffolding problem. The key step in our algorithm is the selection of contig orientations and a set of read pairs consistent with these orientations (and locally consistent with each other) such that the overall likelihood of selected pairs is maximized. As in previous works [75, 84], the likelihood model we employ assumes independence of the HTS read pairs. The currently implemented model takes into account read mapping uncertainty due to overlap with annotated contig repeats as well as variations in contig coverage. The model can be easily extended to incorporate sequencing errors and the distribution of insert lengths; currently we only use the latter to eliminate read pairs with highly discordant insert length lower-bounds and to compute ML estimates for the final gap lengths. Likelihood maximization is formulated as an integer linear program (ILP). Unlike MIP [92], our ILP formulation selects contig orientations and a set of locally consistent read pairs but neither explicitly orders the contigs nor fully guarantees global consistency of selected pairs. The latter are achieved by decomposing the set of selected read pairs into linear paths via bipartite matching.

Scalability of our algorithm, referred to as SILP2, is achieved by adopting a non-serial dynamical programming (NSDP) approach [100]. Rather than solving one large ILP, several smaller ILPs can be solved seperatly and composed to find the complete and optimal solution. The order in which the smaller ILPs are solved is determined by the 3-connected components of the underlying scaffolding graph, which can be efficiently identified in linear time via the SPQR-tree data structure [42, 36].

Compared to the preliminary version of the algorithm published in [63], referred to as SILP1, SILP2 is based on explicit formalization of likelihood maximization as the optimization objective. We present experiments with several likelihood models capturing read mapping uncertainty and/or non-uniformity of contig coverage. SILP2 also achieves higher scalability by using a more efficient NSDP algorithm than SILP1. This greatly reduces the need for heuristics such as the hierarchical scaffolding approach of SILP1, whereby scaffolding is performed by progressively decreasing the minimum bound on the size of read pair bundles. We have also expanded the set of experiments to include scaffolding of low-complexity metagenomic samples. The results show that SILP2 compares favorably to previous methods OPERA and MIP in both scalability and accuracy for scaffolding single genomes of up to human size, and significantly outperforms them on scaffolding low-complexity metagenomic samples.

1.2 Methods

Given a set of contigs C and a set of read pairs R, the scaffolding problem asks for the most likely orientation of the contigs along with a partition of the contigs into ordered sets connected by read pairs of R. The main steps of the SILP2 algorithm are as follows (see Figure 1-1 for a high-level flowchart). We first map the read onto contigs using Bowtie2 [54], disregarding pairing information in the mapping process. Alignments are processed to extract read pairs for which both reads have unique alignments, and the alignments are onto distinct contigs. A scaffolding graph is then constructed with nodes corresponding to contigs and edges corresponding to extracted read pairs. The scaffolding graph is partitioned into 3-connected components using the SPQR tree data structure [42, 36] implemented in OGDF [20]. The maximum-likelihood contig orientation is formulated as an ILP that is efficiently solved by applying non-serial dynamic programming based the SPQR tree data structure. Next, scaffold chains are extracted from the ILP solution by using bipartite matching and breaking remaining cycles. Finally, maximum likelihood estimates for the gap lengths are obtained using quadratic programming. Below we detail the key steps of the algorithm, including scaffolding graph construction, the maximum likelihood models used for contig orientation and mapped read pair probability estimation, then we briefly overview the orientation, the ILP formulation and the improved NSDP algorithm for efficiently solving the ILP.

Scaffolding graph. The scaffolding problem is modeled with a *scaffolding graph* $G = (V, E)$, where each node $i \in V$ represents a contig and each edge $(i, j) \in E$ represents all read pairs whose two individual reads are mapped to the contigs i and j, respectively. Each read in a pair is aligned either to the forward or reverse strand of corresponding contig sequence, and this results in 4 possible configurations for a read pair (denoted A, B, C, or D, see Figure 1-2) which can be modeled as a bidirected edge [92, 26, 63]. Orientation of contigs and the bidirected orientation of edges should agree (be concordant) with each other and should not result in any directed cycles for linear genomes (e.g. eukaryotes).

Maximum likelihood scaffold graph orientation. As an intermediate step towards solving the scaffolding problem, we consider the problem of determining an orientation of the scaffolding graph, which includes choosing one of the two possible orientations for each node (contig) $i \in V$ as well as choosing for each edge $(i, j) \in E$ one of the four bidirections that is concordant with the orientations of i and j. A

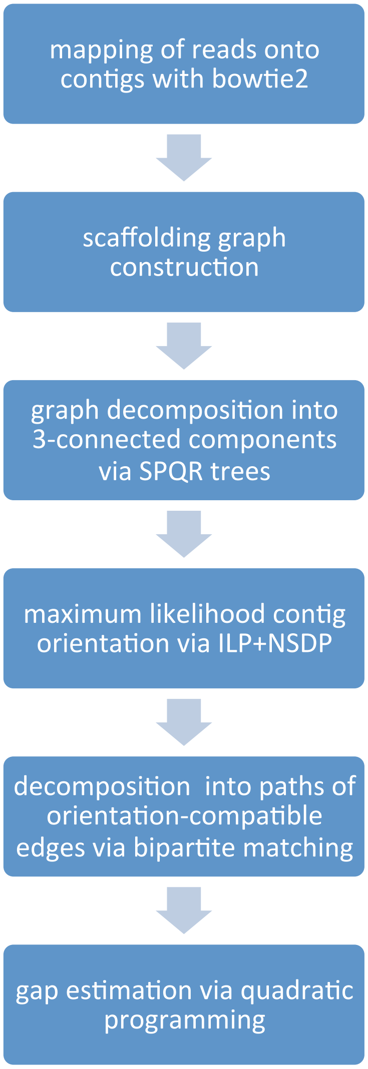

mapping of reads onto contigs with bowtie2
scaffolding graph construction
graph decomposition into 3-connected components via SPQR trees
maximum likelihood contig orientation via ILP+NSDP
decomposition into paths of orientation-compatible edges via bipartite matching
gap estimation via quadratic programming

Figure 1-1: A flowchart describing the SILP workflow.

common way to reduce an inference problem to an optimization problem is to seek a feasible solution with maximum likelihood. Let each observation, i.e., aligned read pair $r \in R$, have a probability p_r of being correct. Any feasible contig orientation $O = O(C)$ either agrees or disagrees with the read pair r. Let R_O be the set of read pairs agreeing with O. Assuming independence of observations, the likelihood of an orientation O can be written as

$$\prod_{r \in R_O} p_r \prod_{r \in R - R_O} (1 - p_r) = \prod_{r \in R} (1 - p_k) \prod_{r \in R_O} \left(\frac{p_r}{1 - p_r} \right)$$

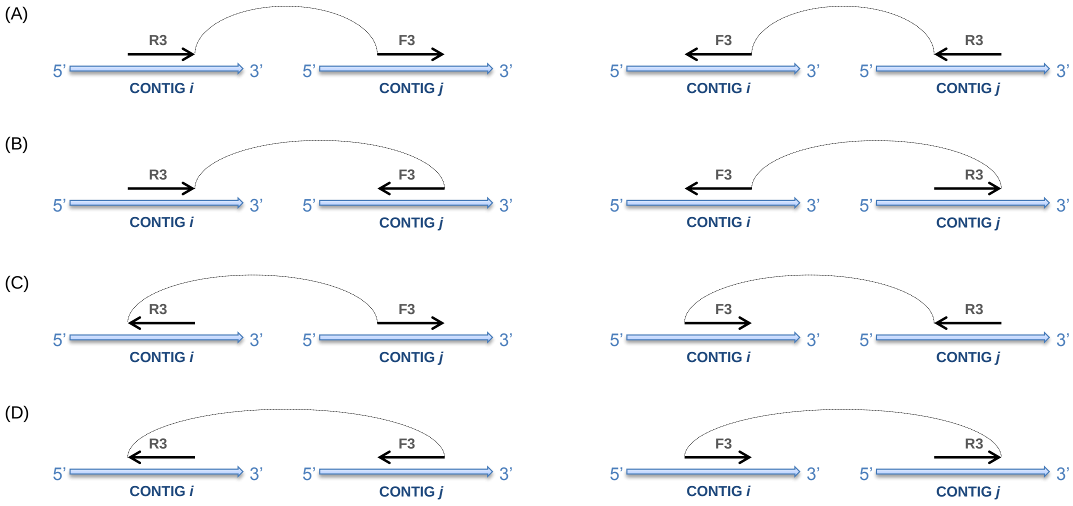


Figure 1-2: The four possible orientations of a read mate-pair linking two contigs i and j.

and hence its log-likelihood is $\sum_{r \in R} \ln(1-p_r) + \sum_{r \in R_O} \ln(\frac{p_r}{1-p_r})$. Since the first sum does not depend on the orientation O, maximizing the log-likelihood is equivalent to maximizing

$$\sum_{r \in R_O} \ln \left(\frac{p_r}{1-p_r} \right) \tag{1.1}$$

over all contig orientations O.

Mapping probability estimation. If p_r's are assumed to be the same for all read pairs, then the objective (1.1) reduces to maximization of the number of read pairs that agree with the contig orientation O. We consider the following factors that reduce the probability p_r that read pair r is aligned correctly:

1. *Overlap with repeats.* As noted above, only pairs for which both reads map uniquely to the set of contigs are used for scaffolding. Still, a read that fully or partially overlaps a genomic repeat may be uniquely mapped to the incorrect location in case repeat copies are collapsed. We preprocess contigs to annotate repeats from known repeat families and by recording the location of multimapped reads. An estimate of the repeat-based mapping probability p_r^{rep} is found by taking the percentage of bases of r aligned to non-repetitive portions of the contigs.

2. *Contig coverage dissimilarity.* Although sequencing coverage can have significant departures from uniformity due to biases introduced in library preparation and sequencing, the average coverage of adjacent contigs is expected to be similarly affected by such biases (all read alignments, including randomly allocated non-unique alignments, are used for estimating computing average contig coverages). If the two reads of r map to contigs i, respectively j, the coverage-based mapping probability of r, p_r^{cov}, is defined as $1 - |coverage_i - coverage_j|/(coverage_i + coverage_j)$.

Note that factors such as repeat content of the sequenced genome and sequencing depth will determine how informative repeat-based and coverage-based mapping probabilities are. Depending on these factors, either p_r^{rep}, p_r^{cov}, or their product may provide the most accurate estimate for p_r. Mismatches and indels in read alignments, that can be caused by sequencing errors or polymorphisms in the sequenced sample, can easily be incorporated in the estimation of mapping probabilities.

Integer linear program. Our integer linear program maximizes the log-likelihood of scaffold orientation using the following boolean variables:

- a binary variable S_i for each contig i, with S_i equal to 0 if the contig's orientation remains the same and $S_i = 1$ if the contig's orientation is flipped w.r.t. default orientation in the final scaffold.

- a binary variable S_{ij} for each edge $(i, j) \in E$, which equals 0 if none or both ith and jth contigs are flipped, and equals 1 if only one of them is flipped.

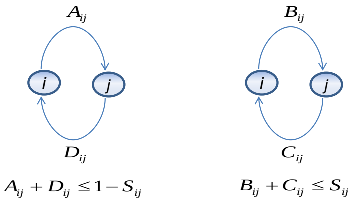
- binary variables A_{ij} (respectively, B_{ij}, C_{ij}, and D_{ij}) which are set to 1 if and only if an edge in state A (respectively, B, C, or D) is used to connect contigs i and j (see Figure 1-3). For any contig pair i and j, at most one of these variables can be one.

Let A_{ij}^r (respectively, B_{ij}^r, C_{ij}^r or D_{ij}^r) denote the set of read pairs supporting state A (respectively, B, C, or D), between the ith and jth contig. Define the constant A_{ij}^w

by

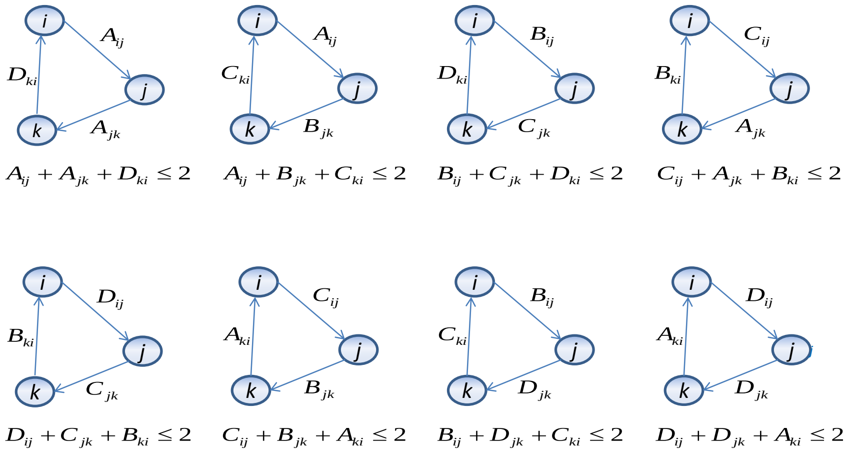
$$A^w_{ij} = \sum_{r \in A^r_{ij}} \ln\left(\frac{p_r}{1-p_r}\right)$$

with B^w_{ij}, C^w_{ij} and D^w_{ij} defined analogously.



A_{ij}
i
j
D_{ij}
$A_{ij} + D_{ij} \leq 1 - S_{ij}$
B_{ij}
i
j
C_{ij}
$B_{ij} + C_{ij} \leq S_{ij}$

Figure 1-3: ILP constraints forbiding 2-cycles.



A_{ij}, A_{jk}, D_{ki}
$A_{ij} + A_{jk} + D_{ki} \leq 2$
A_{ij}, B_{jk}, C_{ki}
$A_{ij} + B_{jk} + C_{ki} \leq 2$
B_{ij}, C_{jk}, D_{ki}
$B_{ij} + C_{jk} + D_{ki} \leq 2$
C_{ij}, A_{jk}, B_{ki}
$C_{ij} + A_{jk} + B_{ki} \leq 2$
D_{ij}, C_{jk}, B_{ki}
$D_{ij} + C_{jk} + B_{ki} \leq 2$
C_{ij}, B_{jk}, A_{ki}
$C_{ij} + B_{jk} + A_{ki} \leq 2$
B_{ij}, D_{jk}, C_{ki}
$B_{ij} + D_{jk} + C_{ki} \leq 2$
D_{ij}, D_{jk}, A_{ki}
$D_{ij} + D_{jk} + A_{ki} \leq 2$
i
j
k

Figure 1-4: ILP constraints forbiding 3-cycles.

We now ready to formulate the ILP for maximizing the log-likelihood of a scaffold orientation:

$$\sum_{(i,j) \in E} (A^w_{ij} \cdot A_{ij} + B^w_{ij} \cdot B_{ij} + C^w_{ij} \cdot C_{ij} + D^w_{ij} \cdot D_{ij}) \tag{1.2}$$

where

$$S_{ij} \leq S_i + S_j \qquad S_{ij} \leq 2 - S_i - S_j \tag{1.3}$$

$$S_{ij} \geq S_j - S_i \qquad S_{ij} \geq S_i - S_j \tag{1.4}$$

$$A_{ij} + D_{ij} \leq 1 - S_{ij} \qquad B_{ij} + C_{ij} \leq S_{ij} \tag{1.5}$$

In this ILP, constraints (1.3-1.5) enforce agreement between contig orientation variables S_i's and edge orientation variables S_{ij}'s, A_{ij}'s, B_{ij}'s, C_{ij}'s, and D_{ij}'s.

Since eukaryotic genomes are linear, a valid scaffold orientation should not contain any cycles. The constraints (1.5) already forbid 2-cycles. Additionally, 3-cycles are forbidden with the constraints shown in Figure 1-4. Larger cycles generated in the ILP solution are broken heuristically because it is infeasible to forbid all of them using explicit constraints.

Non-serial dynamic programming. For large mammalian genomes, the number of variables and constraints is too large for solving the ILP (1.2)-(1.5) via standard solvers (SILP2 uses CPLEX[48] which is available free of charge for academic institutions). We adopt the non-serial dynamic programming (NSDP) paradigm to overcome this barrier and to optimally solve the problem. NSDP is based on the interaction graph with nodes corresponding to ILP variables and edges corresponding to the ILP constraints – two nodes are adjacent in the interaction graph if their associated variables appear together in the same constraint. Through the NSDP process, variables are removed in the way that adjacent vertices can be merged together [100]. The first step in NSDP is identifying weakly connected components of the interaction graph. We find the 2- and 3-connected components of the interaction graph with efficient algorithms and then we solve each component independently in such a way that the solutions can be merged together to find the global solution.

All constraints (1.3-1.5) as well 3-cycle constraints connect S_i's following the edges of the scaffolding graph. Therefore, the S_i-nodes of the interaction graph for our ILP will have the same connectivity structure as the scaffolding graph $G = (V, E)$. As it has been noticed in [26], the scaffolding graph is a bounded-width graph and should be well decomposable in 2- and 3-connected components. The SPQR-tree data structure is employed to determine the decomposition order for 3-connected components the scaffolding graph [42]. The solution to each component of the scaffolding graph is found using a bottom up traversal through which each component is solved 2 times: for similar and opposite orientations of the common nodes. The objective value of each case is then entered into the objective of the parental component. Having the solution of all components, top down DFS starting from the same root is performed to apply the chosen solution for each component.

Below we illustrate the way how the solution is computed in stages through each of which the results of the previous stage are combined to dynamically solve the problem. Obviously, an isolated connected component will not influence other components. Moreover, it has been shown in [92] that 2-connected components can be solved independently. As it can be seen in Figure 1-5(a), after removing the articulation point (1-cut) to decompose the graph into 2-connected components, each component is solved with the same arbitrary direction assigned to the common node, and then the resulting solutions are collapsed into the parent solution. The pre-assigned direction will never affect the parent solution since all contigs in the scaffold can be flipped at the same time.

Still, 2-connected components can be very large, so we look for 2-cuts in order to decompose the graph into significantly smaller 3-connected components. Figure 1-5(b) shows that splitting the two 2-cut nodes i and j decomposes the graph into 3-connected components A and B. The ILP for component A is solved twice to obtain

(1) the ILP solution sol_{00} in which the 2-cut nodes i and j are constrained to both have default orientations;

(2) the ILP solution sol_{01} in which the 2-cut nodes i and j are constrained to have opposite orientations.

The two solutions are combined to solve the ILP for component B. The ILP objective for component B should be updated by adding the term of $sol_{00}+(sol_{01}\text{-}sol_{00})\cdot S_{ij}$ or, equivalently, the value sol_{00} should be added to A_{ij}^{w} and D_{ij}^{w} and the value sol_{01} should be added to B_{ij}^{w} and C_{ij}^{w}. The overall solution is obtained by identifying the common nodes of the components. In the example on Figure 1-5(b), the optimal solution happens when 2-cut nodes have opposite directions. The corresponding solution of ILP for the component A should be incorporated in the overall solution. When the scaffolding graph has 3-connected components too large to handle, 3-cuts could also be used for decomposition.

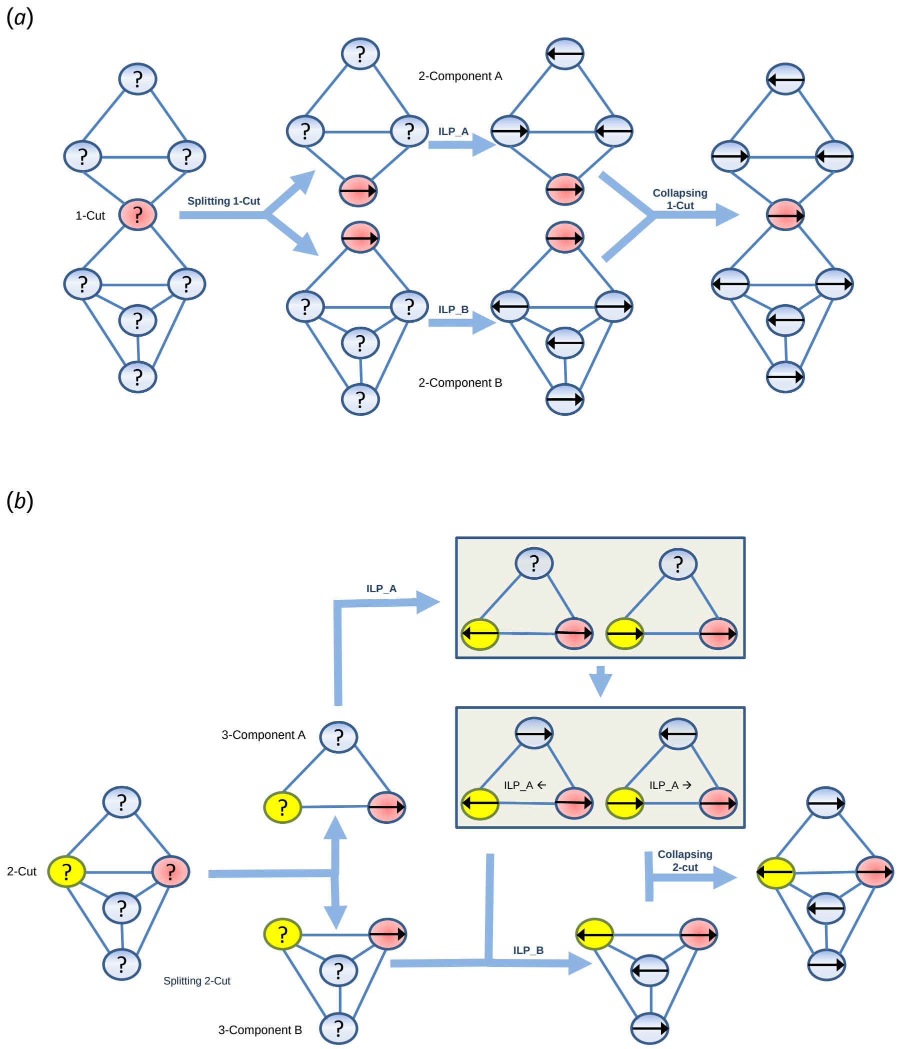


Figure 1-5: *(a)* Graph decomposition into 2-connected components: Red (1-cut) node splits the graph into two 2-connected components A and B. The ILP is solved for each component separately. If the direction of the cut node in the ILP solution for B is opposite to the one in the solution for A, then the solution of B is inverted. Then ILP solutions for A and B are collapsed into the parent solution. ***(b)*** Graph decomposition into two 3-connected components: Red and yellow (2-cut) nodes split the graph into two 3-connected components A and B. The ILP is solved for component A twice – for the same and the opposite directions assigned to two 2-cut nodes. Then these two solutions are used in the objective for the ILP of component B. Finally, ILP solutions for A and B are collapsed into the parent solution.

The pseudo-code of the SILP2 NSDP algorithm for processing 3-connected components is given in Figure 1-5. SILP2 is different from SILP1 in the else clause – instead of solving ILP for each of four possible combinations of assignments for S_i and S_j as in SILP1, ILP is solved only two times for combinations $S_i = 0$ & $S_j = 0$ and $S_i = 0$ & $S_j = 1$.

Thinning Heuristic. Unfortunatly the largest tri-connected component may still induce an ILP too large for CPLEX to solve in a reasonable amount of time. In order to address this problem a thinning heuristic is applied to the scaffolding graph. This scenario can be detected by setting a threshold on the maximum number of contigs allowed in a tri-connected component. When a component exceeds the threshold the number of read pairs necessary to induce an edge is increased by one and decomposition recomputed until there is no component above the threshold.

1.3 Results and Discussion

1.3.1 Datasets and Quality Measures

In order to asses the quality and scalability of our scaffolding tool we developed a testing framework which closely mimics real world scaffolding problems. We utilized the Staphylococcus aureus (staph), Rhodobacter sphaeroides (rhodo) genomes and chromosome 14 of HapMap individual NA12878 (chr14) from the GAGE [93] assembly comparison. Finally, in a test case designed to stress scalability, contigs from a draft assembly of individual NA12878 (NA) created by [103] were scaffolded using short-read data.

In all test cases the read pairs used for scaffolding are aligned against the contigs using bowtie2 [54]. Each read in a pair was required to be aligned uniquely according to the default scoring scheme, for the pair to be considered valid. Each scaffolder was given the same set of valid read pairs. Two of the leading external scaffolding tools MIP [92] and OPERA [26] are used in this comparison. Although many other tools do exist, these two are widely utilized and actively maintained.

The three small test cases are used to test both correctness and scalability of the scaffolding tool. In order to test correctness, contigs simulating a draft assembly were created by placing gaps in the genome. The contig and gap sizes were sampled uniformly at random from the collection of all the assemblies used in the GAGE comparison. The procedure to generate the contigs was to alternatively sample with replacement from the set of all contig sizes, and gap sizes. In this way a simulated scaffold can be generated so that the position and relative orientation of all contigs and all gap sizes are known. The orientation of the simulated contigs was randomized to prevent biases.

For each genome 10 replicates were created, all subsequent results are the average of the 10 replicates. By creating simulated contigs with no assembly error, the accuracy of subsequent scaffolds can be evaluated exactly. Although the contigs were simulated, real read pairs were aligned against them and used as input. Table S1 in Additional File 1 describes the characteristics of each dataset.

The NA12878 test case was produced by simply using the contigs created in the SGA [103] assembler publication. The read pairs were obtained from a different lab, however they were generated using the same biological source material (ERP002490). Although more read pairs were available a random subset of approximately 2x coverage was used.

Finally a simulated metagenomics test case was created to explore the feasibility of utilizing SILP2 to scaffold metagenomes. This was created by artificially mixing the staph and rhodo contigs and reads at varying proportions.

A natural and common parameter present in all scaffolding algorithms in the bundle size, or the number of read pairs spanning two contigs. This parameter is a natural control of sensitivity and specificity; requiring more support increases specificity at the price of sensitivity and vice-versa. It should be noted that every scaffolding tool tested, including SILP2 does not abide by the set parameter absolutely. Each method raises it in order to ensure efficient operation. The simulated test cases were evaluated at several bundle sizes to asses its effect on accuracy and scalability. The NA12878 test case was only evaluated at the minimum feasible value due to resource constraints.

1.3.2 Accuracy

Calculating the accuracy of de novo assemblies or scaffolds is quite difficult. One of the key challenges is deciding on the appropriate measure. In this comparison we elect to present several metrics which will likely have different weight depending on the background and intention of the reader.

For the simulated contigs we treat scaffolding as a binary classification problem where methods attempt to predict true adjacencies in the test dataset. The accuracy and sensitivity can be directly measured by computing true positive, and false positive rates. One common summary is MCC, or *Mathews Correlation coefficient*. This measure assess sensitivity and specificity simultaneously. In the context of scaffolding, this measure illustrates how many correctly ordered and oriented scaffolds were created.

An alternative measure, commonly utilized in genome assembly comparison publications [93, 35] is the notion of corrected N50. Where N50 is the weighted mean scaffold size, the corrected N50 is the same statistic after errors are removed. This can be computed exactly on simulated data, however an alignment based approximation must be used on real test cases.

Finally the usefulness of a genome can also be measured by the number of identifiable biological features captured. Here we capture this measure by recording the percentage of known genes that are found contiguous in the scaffolds.

MCC

The MCC metric, as seen in Figure 1-6, indicates that SILP2 is able to correctly join the most contigs, followed by OPERA and finally MIP. This order holds for all three simulated test cases. Interestingly all three methods see a decrease in MCC on staph, but some have increases on rhodo and chr14. This trend illustrates the difficult to define variables such as genome uniqueness, assembly and read error which can make

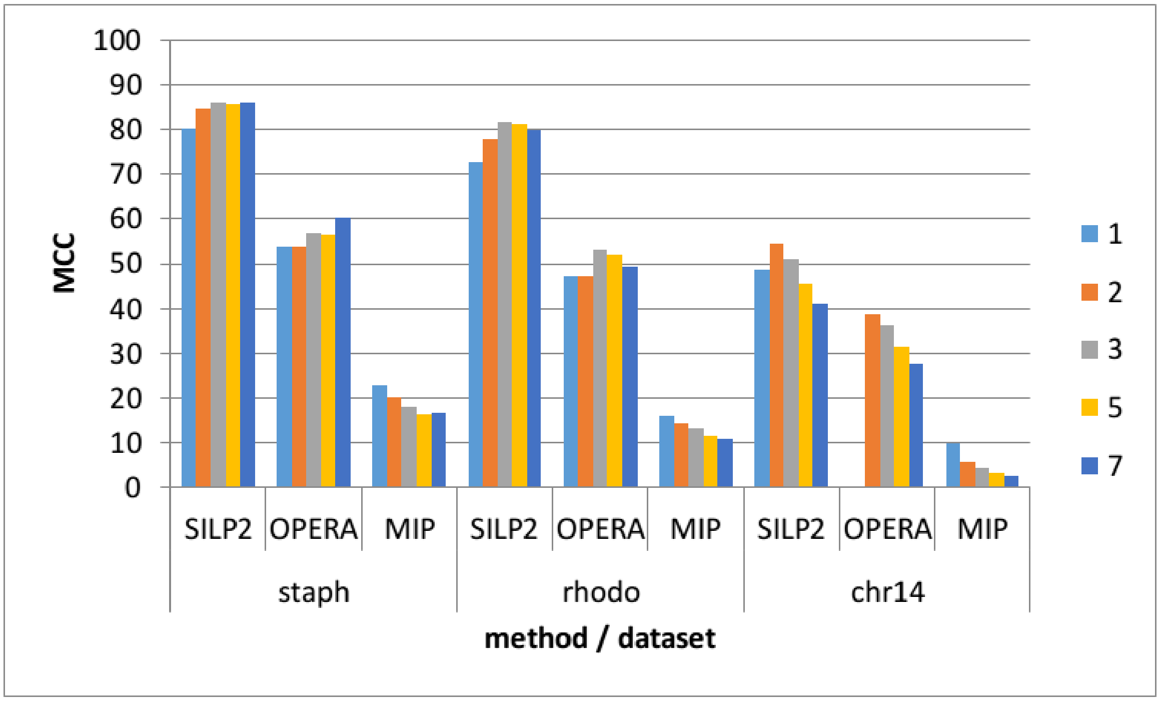


Figure 1-6: MCC for SILP2, OPERA and MIP across bundle sizes and all three simulated assemblies staph, rhodo, chr14. Note at bundle size 1 for chr14 OPERA exceeded the allowed runtime of 2 days and did not complete.

smaller genomes more challenging that larger genomes.

While MCC is natural to a computer scientist its useful to a biologist is lacking because the content of the contigs is ignored. A biologist typically asses a scaffold by the N50, Unfortunately this measure does not reflect the accuracy of the scaffolds and rewards aggressive merging. Using MCC or its constituent components as metrics gives greater clarity to the researcher comparing different tools.

N50

The most common metric found in genome assembly and scaffolding is N50. The most recent iteration of benchmark projects have transformed this descriptive number into an accuracy measure by introducing alignment based corrections. Here the scaffolds are aligned against a reference and miss-alignments are interpreted as orientation, or placement errors. We have developed a more efficient implementation of the correcting method developed by [35]. This enables the tool to be utilized on the NA12878 test case at the cost of accuracy.

The true N50 value can be determined when using simulated contigs by breaking incorrect scaffolds, this measure is denoted as TPN50. An analog to the TPN50 measure can be obtained by aligning the scaffolds against the known reference. Scaffolds (and contigs) are broken at mis-assembled or mis-scaffolded regions. This post-alignment metrics can be obtained from the assembly evaluation tool called QUAST [35] and it is denoted as NA50.

Unfortunately the implementation of QUAST required more than 128GB of RAM to evaluate the NA12878 test case, and therefore could not be run. We wrote an alternative implementation of NA50 called ALN50 which is more efficient, but follows a similar framework. Both NA50 and ALN50 are found in Figure 1-7. Although NA50 and ALN50 do not agree, they do indicate similar trends between methods. Therefore ALN50 will be used henceforth. In the staph genome, OPERA is clearly the best performing tool, followed by SILP2 and then MIP. However on the rhodo genome, SILP2 performs best, followed by OPERA then MIP.

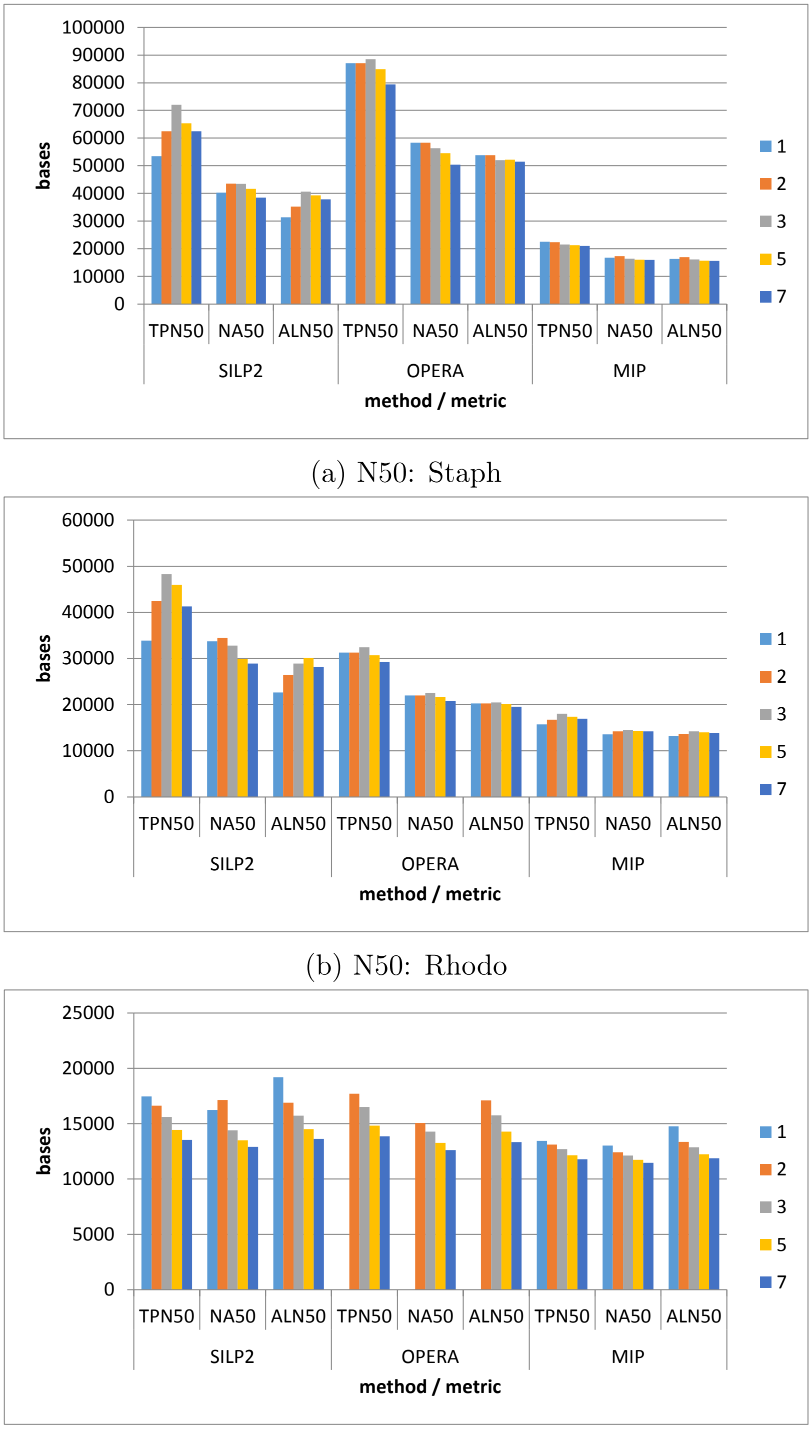
First the highest ALN50 is always found at bundle size 3 or 5. If the intent of the assembly is to maximize N50 then clearly no algorithm should be run with bundle size less than 3. However, as it was pointed out in both GAGE and QUAST [35, 93], N50 is a misleading metric and alternative measures may be a better judge.

Additionally it can be seen that both OPERA and SILP2 have approximately the same TPN50 in the staph and chr14 test cases, however in rhodo, SILP2 clearly outperforms OPERA and MIP at all bundle sizes. It is not clear why SILP2 performs much better on rhodo, and approximately equivalent on the others.

For the complete genome SILP2, OPERA and MIP reported an N50 of 26,235, 39,366, 26,235 respectively. This is consistent with the observations from the synthetic data sets.

Gene Reconstruction

An alternative measure of the completeness of a scaffold is the number of genes aligned against the scaffold. For a given percentage of completeness the number of



(a) N50: Staph

(b) N50: Rhodo

(c) N50: Chr14

Figure 1-7: TPN50 is obtained by breaking incorrect scaffolds, ALN50 is the post-alignment metric developed by us, and NA50 is the QUAST equivalent. The colors indicated in the legend correspond to the bundle size 1 through 7. OPERA was unable to complete on bundle size 1 for chr14 dataset.

Table 1.1: In order for a gene to be considered reconstructed 90% of its sequence must be found in a contiguous scaffold. Dashes (-) indicate the method was unable to complete and therefore the gene count could not be computed.

genome	bundle	SILP2	OPERA	MIP	total
staph	1	1,727.70	1,168.50	1,545.00	2692
	2	1,727.70	1,168.50	1,559.50	
	3	1,727.70	1,210.60	1,575.30	
	5	1,727.70	1,262.70	1,584.60	
	7	1,727.40	1,280.40	1,588.50	
rhodo	1	2022.7	1618.6	1897.3	3067
	2	2022.7	1618.6	1907	
	3	2022.6	1751.1	1894	
	5	2022.6	1834.2	1921.3	
	7	2022.6	1853.3	1933.3	
chr14	1	350.9	-	349.6	529
	2	352.00	330.10	350.40	
	3	352.40	336.90	350.40	
	5	352.40	337.50	351.70	
	7	352.40	337.60	3.00	
NA12878 2x	1	30817	-	30817	34039
	2	30850	30809	30849	

genes found in the corrected scaffold is an indicator of the usefulness of the genome.

As seen in Table 1.1, SILP2 almost consistently equals or outperforms both OPERA and MIP at all bundle sizes and for each genome. The difference between SILP2 and MIP is often quite small.

Runtime

One key advantage of SILP2 over other scaffolding tools is its speed and scalability. Table 1.2 gives the runtime of SILP1, SILP2, OPERA and MIP on single-genome testcases. All experiments were conducted on a Dell PowerEdge R815 server with quad 2.5GHz 16-core AMD Opteron 6380 processors and 256Gb RAM running under Ubuntu 12.04 LTS. IBM ILOG CPLEX 12.5.0.0 was used as ILP solver through the CPLEX Python API. Reported runtimes are only for the scaffolding portion of each program. Read alignment and pre-processing steps are not included, but it was observed that all methods had comparable pre-processing times.

Table 1.2: All timing was captured only during the scaffolding phase of each tool, all read alignment and formatting procedures were excluded from timing. The number is the average of 10 runs for each genome. A dash (-) indicated the tool was unable to complete in the allotted time of 2 days for staph,rhodo,chr14 and 3 days for NA12878.

genome	**bundle**	**SILP1**	**SILP2**	**OPERA**	**MIP**
staph	1	1237	6.4	2538.1	35.8
	2	738	4.5	1456.5	17
	3	305	4	878.5	12.834
	5	142	3.9	386.9	10.54
	7	51	4.3	241	10.115
rhodo	1	1134	10	2297	118.953
	2	632	4.1	455.2	25.3
	3	486	3.6	5.7	10.995
	5	86	3.4	2	8.778
	7	75	3	1.6	8.217
chr14	1	-	64.7	-	706.3
	2	-	27.6	99.25	189.685
	3	629	25.5	11	137.67
	5	370	21.5	12	107.85
	7	400	19.25	10.75	94.9875
NA12878 2x	1	-	55.2	-	89.3
	2	-	1670	76.49	53.28
	3	37751	3878	7875	121.61
	5	27341	3183	4270	134.6
	7	27470	3626	2180	125.66

On the staph, rhodo and chr14 datasets, it was observed that SILP2 was quicker at higher bundle sizes and no worse than OPERA or MIP at lower bundle sizes. The NA12878 testcase was extremely challenging for all methods and demonstrated the effect of heuristics on large test cases. It is clear from the reduced runtimes that all 3 methods activate some sort of heuristic at lower bundle sizes. The difference between SILP1 and SILP2 is evident at all bundle sizes.

The NA12878 genome was also scaffolded by SILP2 using 20x coverage reads, with a runtime of 18,205 seconds at bundle size 1. Negligable improvement in accuracy over the 2x dataset was observed. From Table 1.2 it is clear that runtime increases with the complexity of the genome more so than the number of read pairs.

1.3.3 Metagenomics

Metagenomics is the study of genetic material recovered from heterogeneous mixtures often found in nature. Just like in the de novo assembly of a single genome, the accuracy and size of the scaffolds is critical to subsequent analysis steps. Our ILP based solution is flexible enough to include new constraints and objectives to better serve this challenging scenario.

In order to test this hypothesis a simulated metagenomic dataset was created utilizing the staph and rhodo genomes from the GAGE dataset. The simulated contigs used previously were mixed, and both sets of reads were aligned with varying fractions (1.0, 9.5 0.25, 0.0) of staph reads present.

Again all three of the major scaffolding tools were tested, however additional weighting scenarios were implemented in SILP2.

The runtime, MCC, SCFN50, TPN50 and ALN50 metrics are detailed for each of the compared methods in Table 1.3. Also an additional scaffolding tool BAMBUS2 [52] was added to the comparison because it was previously shown to work well in the metagenomic scaffolding context.

Interestingly all SILP2 variants fare much better than both OPERA, MIP and BAMBUS2 even with no staph reads present (this differs from results in Figure 1-7 because the rhodo reads were aligned to both staph and rhodo contigs). It is unclear is the different methodology used in SILP2 sets it apart, or if an implementation quirk throws off the other scaffolders. However across all metrics SILP2 variants perform the best.

In both SILP2 variants and MIP it is observed that the TPN50 decreases as fewer staph reads are utilized. This is expected since there are fewer opportunities to connect staph contigs and both staph and rhodo contigs are used in the calculation of N50. There is no major differences between the variants of SILP2. The coverage based weight seems to improve MCC at the cost of a slightly lowered TPN50 when compared to no weights.

This highly simplified test scenario is not designed to fully explore metagenomic scaffolding, rather to point out an opportunity to further external genome scaffolding algorithms.

1.4 Conclusions

Scaffolding in an important step in the de novo assembly pipeline. Biologists rely on an accurate scaffold to perform many types of analysis. The larger the scaffold the more useful it will be to them. Recent advances in de novo assemblers has made it feasible to create draft assemblies for large mammalian genomes. We believe that SILP2, coupled with the most recent scalable assemblers will produce the largest and most complete assemblies. This is made possible utilizing non-serial dynamic programming approach to solve our robust ILP. The ILP formulation for the maximum likelihood model is shown to be flexible enough to handle metagenomic samples.

The future work includes more thorough experimental validation of SILP2 and comparison BAMBUS2 [52] on metagenomic samples. Also we are going to validate SILP2 using the methodology and benchmarks from the recently published comparative study [45].

Table 1.3: The second column indicates the percentage of total read pairs used from the staph genome testcase, all of the rhodo pairs were used. SCFN50 is the uncorrected N50 reported by each scaffolding tool. The N50 of the contigs alone is 10,339bp. The integer appended to SILP2 indicates the bundle weight; 0: none, 1: coverage, 2: repeat, 3:coverage * repeat. All methods were run at bundle size 1, the reported number is the average of 10 runs except for OPERA where 6 of the test cases exceeded runtime limits. TPN50 and MCC were unable to be computed for BAMBUS2 because the generated AGP had a non-standard format.

METHOD	FRAC STAPH	RUNTIME	SCFN50	ALN50	TPN50	MCC
SILP2_0	1.00	14.3	51,775.0	20,647	34,495	67.6
SILP2_0	0.50	13.3	50,450.0	20,103	36,356	69.1
SILP2_0	0.25	12.7	47,731.0	20,761	35,323	69.3
SILP2_0	0.00	11.5	21,753.0	13,948	15,649	39.9
SILP2_1	1.00	14.3	52,557.0	20,655	33,683	67.5
SILP2_1	0.50	13.8	48,701.0	20,337	35,750	69.0
SILP2_1	0.25	13.4	49,766.0	20,752	35,146	69.0
SILP2_1	0.00	11.0	21,925.0	13,847	15,511	39.3
SILP2_2	1.00	14.3	43,144.0	20,631	31,160	66.3
SILP2_2	0.50	13.5	42,244.0	20,198	32,477	67.5
SILP2_2	0.25	13.2	45,137.0	21,161	31,562	67.6
SILP2_2	0.00	10.8	22,190.0	13,813	16,205	41.6
SILP2_3	1.00	14.1	43,646.0	19,998	28,856	65.3
SILP2_3	0.50	13.2	41,893.0	19,790	30,504	66.6
SILP2_3	0.25	13.0	42,188.0	19,945	30,449	66.4
SILP2_3	0.00	11.5	21,820.0	13,781	15,635	40.0
OPERA	1.00	2247.2	15,573.0	13,082	10,386	10.1
OPERA	0.50	1567.6	13,928.0	12,006	10,440	10.7
OPERA	0.25	884.0	14,786.0	12,617	10,507	10.5
OPERA	0.00	544.3	11,121.0	10,720	10,273	4.9
MIP	1.00	129.9	20,104.0	12,861	18,672	18.4
MIP	0.50	121.3	19,807.0	12,488	17,613	17.4
MIP	0.25	114.0	18,520.0	12,269	16,680	17.2
MIP	0.00	114.1	12,690.0	10,894	12,434	8.7
BAMBUS2	1.00	1025.89	11,251.0	11,238	-	-
BAMBUS2	0.50	1452.75	10,781.0	10,822	-	-
BAMBUS2	0.25	1676.75	10,806.0	10,834	-	-
BAMBUS2	0.00	2272	11,526.0	11,698	-	-

Chapter 2

Biomarker Selection and Predictive Modeling

There is an ever-expanding range of technologies that generate very large numbers of biomarkers for research and clinical applications.[1] Choosing the most informative biomarkers from a high-dimensional data set, combined with identifying the most reliable and accurate classification algorithms to use with that biomarker set, can be a daunting task. Existing surveys of feature selection and classification algorithms typically focus on a single data type, such as gene expression micro-arrays, and rarely explore the model's performance across multiple biological data types.

This paper presents the results of a large scale empirical study whereby a large number of popular feature selection and classification algorithms are used to identify the tissue of origin for the NCI-60 cancer cell lines. A computational pipeline was implemented to optimally tune and evaluate the performance of each pair of feature selection and classification methods on five different data types available for the NCI-60 cell lines in models exploiting both large and small numbers of biomarkers.

As expected, the data type and number of biomarkers have a significant effect on the performance of the predictive models. Although no model or data type uniformly outperforms the others across the entire range of tested numbers of markers, several

[1] The results presented in this chapter are based on joint work with E. Hemphill, C. Lee, I.I. Mandoiu, and C.E. Nelson published in [40].

clear trends are visible. At low numbers of biomarkers gene and protein expression data types are able to differentiate between cancer cell lines significantly better than the other three data types, namely SNP, array comparative genome hybridization (aCGH), and microRNA data. Interestingly, as the number of selected biomarkers increases best performing classifiers based on SNP data match or slightly outperform those based on gene and protein expression, while those based on aCGH and microRNA data continue to perform the worst. It is observed that one class of feature selection and classifier are consistently top performers across data types and number of markers, suggesting that well performing feature-selection/classifier pairings are likely to be robust in biological classification problems regardless of the data type used in the analysis.

2.1 Motivation

Due to the recent rise of big-data in biology, predictive models based on small panels of biomarkers are becoming increasingly important in clinical, translational and basic biomedical research. In clinical applications such predictive models are increasingly being used for diagnosis [1], patient stratification [44], prognosis [79], and treatment response, among others.

Many types of biological data can be used to identify informative biomarker panels. Common ones include micro-array based gene expression, microRNA, genomic copy number, and SNP data, but the rise of new technologies including high-throughput transcriptome sequencing (RNA-Seq) and mass spectrometry will continue to increase the diversity of biomarker types readily available for biomarker mining.

Useful predictive models are typically restricted to use a small number of biomarkers that can be cost-effectively assayed in the lab [23]. The use of few biomarkers also reduces the effects of over-fitting, particularly for limited amounts of training data [72]. Once training data has been collected and appropriate procedures for normalization of primary data have been defined, assembling a robust biomarker panel requires the solution of two main computational problems: *feature selection*, to identify a short list of informative biomarkers, and *classification*, used to make predictions for new samples based on patterns extracted from the training data. Both of these steps have been explored extensively in the statistics and machine learning literature, and many alternative algorithms are available for each. Due to the sheer number of available choices and the lack of predictable interactions between feature selection method, classification algorithm, and data type, assembling the most robust biomarker assay for a given biomedical application is rarely undertaken systematically. Rather, it is more often driven by the intuition and a priori preferences of the statistician.

Available feature selection methods can be grouped into three broad categories: filter, wrapper and embedded. Filtering approaches use an easy to calculate metric which allows quick ranking of the features, with top ranking features being selected. Wrapper methods use a classification algorithm to interrogate the effect of various biomarker subsets. Embedded approaches are classification algorithms which eliminate features as part of the training process. Recent studies [39, 53, 55] investigated the influence of feature selection algorithms on the performance of predictive models and provided a framework for thorough comparison of approaches. However the effect of the number of biomarkers selected and high-dimensional data type was not explored.

There are hundreds of publications describing classification algorithms and their applications to genetic research and medicine. Many publications advocating a new method employ a limited comparison between similar approaches. However non-uniform validation strategies make it difficult to assess performance of a wide variety of approaches. A previous study compared both classification and feature selection approaches in a unified framework [55], however the effect of biological data type was not explored, but it was observed that the biological question does have an effect on the best model. Additionally most comparisons typically overlook the effect of model parametrization even though the choice of parameters can have profound effects on performance.

This work presents a large scale empirical comparison of the effects of the the interaction between the main components of the predictive model (i.e., feature selection and classification algorithms), the number of features utilized, and the underlying data type on the performance of the overall model. This study also implements exhaustive parametrization of all models to ensure a fair comparison between models.

In order to test the performance of the large number of models tested in this study, and in order to be able to run direct comparisons of the models on different biological data types, we took advantage of the publicly available NCI-60 cancer cell line data set [85]. The NCI-60 cell line collection represents a carefully curated collection of 60 independent cancer cell lines derived from nine types of cancer occurring in 60 individual patients. Each line has been uniformly cultured and DNA fingerprinted to ensure independence [67]. In addition, the NCI-60 cell lines have been subjected to extensive molecular characterization including mRNA microarray [99], microRNA [31], protein lysate arrays [99], SNP arrays [94], and aCGH analysis [109]. For these reasons, the NCI-60 data set represents a tremendous research tool for exploring and benchmarking Omics-type approaches to cancer classification and therapeutics.

Cancers are widely believed to derive from a single event in which one cell escapes the many surveillance mechanisms in place to prevent uncontrolled proliferation. Once this has occurred, the cancer often evolves quickly, rapidly acquiring large numbers of mutations, ranging from small point mutations to very large chromosomal aberrations and regional amplifications (DNA duplications). The original identity of the cancer cell (its cell type or tissue type) appears to exert a very strong influence on the course of evolution of the cancer. For this reason, characteristic mutations will often be found in cancers derived from the same tissue, even in different patients. In addition, because identical cell types from different patients will share very similar gene expression signatures, cancers derived from these tissues will often do the same. In the present study we take advantage of these two features of cancer to test the ability of various statistical models to correctly infer the cell type (or “tissue-of-origin”) of each cancer cell line. The ability to make this inference correctly not only represents an excellent test of these models on real biological data, it is a good example of the type of classification ability required for targeted cancer therapeutics.

2.1.1 Feature selection without replicates

In certain scenarious the biological dataset may contain no replicates, missing observation and binary information. A new feature selection approach utilizing a robust ILP that is designed to work with manually curated cell type specific expression data from the Stem Cell Lineage Database [41]. Also this particular data type has a lineage or hierarchy defined based on the development of each cell type. Therefore this work introduces a modified accuracy measure that is better suited for this scenario.

Typical notation for the biomarker selection problem is, given n cell types with associated p-marker expression profiles, it is desirable to find a subset of markers that allows us to distinguish one cell type from another. This can be regarded as a supervised feature selection problem [37], where each cell type forms a class of one instance and the goal is to find a subset of markers achieving high classification accuracy. However, due to the sparseness of the expression data in this context, standard feature selection algorithms are not applicable. Let $E = (E_{ij})$ be the $n \times p$ expression matrix where $E_{ij} \in \{-1, 0, 1\}$ and -1,0,1 denote that marker j is absent, unknown or present in cell type i respectively. We denote by $D_j(i_1, i_2)$ the distance between cell types i_1 and i_2 indexed by marker j. The distance is 0 if marker i_1 or i_2 is unknown or the expression is the same, 1 otherwise.

This problem is very similar to the classical minimum set covering problem (MSCP) [78]. In the context of the MSCP, there are p sets and set $S_j = \{(i_1, i_2)|i_1 < i_2 \text{ and } D_j(i_1, i_2) = 1\}$. The goal is to find a smallest collection of sets, C, such that $\bigcup_{S \in C} S = \{(i_1, i_2)|i_1 < i_2\}$. Table 2.1 shows an example 3-marker expression profiles of 4 cell types. Based on this expression matrix, the 3 sets are listed in table 2.2, where 1 (0) denotes presence (absence) of a pair in a set. Set S_i contains cell type pairs that are separable by marker i. We can see that S_1, S_2 and S_2, S_3 are two smallest collections of sets covering all the pairs. Although MSCP is known to be NP-hard [50], it is still feasible to find an exact solution to this problem using ILP since the number of informative markers will be small.

The size of the chosen subset, denoted by θ, is typically not fixed a priori. However

	Marker		
Cell Type	1	2	3
1	1	1	0
2	0	-1	0
3	1	1	0
4	0	0	1

Table 2.1: Example marker profile for 4 cell types and 3 markers. Typically 1 is present, -1 absent and 0 is unknown.

	Set		
Cell Type Pair	S1	S2	S3
(1, 2)	1	1	0
(1, 3)	0	1	0
(1, 4)	1	1	1
(2, 3)	1	1	0
(2, 4)	0	1	1
(3, 4)	1	0	1

Table 2.2: The sets induced by table 2.1

it is assumed that the size of θ is directly correlated to the cost of the resulting assay. Therefore not only should the subset *theta* be maximally informative it should also have minimal size and therefore cost. Typically the actual number of biomarkers are fixed to certain sizes dictated by the physical format of the assay.

Finally while many biological questions are binary, or two-class, the prevalence and relatively low cost of obtaining of high-throughput data has allowed for much more complex questions to be asked. In the developmental context being explored here there can be dozens of classes, which actually fall into a hierarchy. Typical accuracy metrics such as the area under the receiver operator curve (AUROC) do not utilize the hierarchical structure available. Intuitively miss classifying a sample as its developmental neighbor should be penalized less than if it were classified in a completely different lineage. Therefore this work explores a modification of AUROC that is more appropriate when a cell type lineage is available.

2.2 Methods

NCI-60 cancer cell-line dataset

In order to test the predictive models in this study we use publicly available data from the NCI-60 cancer cell lines as provided by CellMiner [85]. For the purpose of this study, we analyzed cancers with at least 5 representative cell lines derived from the same tissue-of-origin (5-9 cell lines per tissue-of-origin). These lines represent cancers emerging from eight tissues: breast, central nervous system, colon, leukemia, melanoma, non-small cell lung, ovarian, and renal cancers. The data types used in this study are gene expression (mRNA) and protein lysate (protein) arrays [99], microRNA [31], SNP arrays [94], and array comparative genome hybridization (aCGH) [109]. All data has been normalized according to best practices for each assay platform prior to downloading for this study [85]. The specific cell lines and data files used in this study can be found in Tables 2.6 and 2.7.

Feature selection methods

The area of feature selection in machine learning has recently been quite robust. There are numerous specialized feature selection algorithms which identify the most informative biomarkers from high-dimensional data. This study utilized at least one approach from each of the three broad categories identified above (filter, wrapper, and embedded). Every approach utilized allowed for a specific number of features to be chosen. No requirement was established that induced a relationship between feature sets from the same algorithm. So the 16 features chosen by one approach are not required to be a subset of the 32 features chosen by the same. For all algorithms we used the implementations in the Scikit-learn [81] Python package, please refer to its associated documentation for specific implementation details.

The fastest and most simplistic selection method is univariate filtering. These approaches rank features according to some score, and the user selects the best k features accordingly. Here the F-statistic (Anova), a generalization of the t-test, is used as a filter, as suggested in [55] and [39]. There are no parameters for this feature selection method.

Wrapper approaches typically use some type of greedy strategy to select influential features using a black box classifier. They are more computationally intensive, however SVM recursive feature elimination (SVM-RFE) is extensively used in medical applications [38]. The parameters considered were the penalty parameter and loss function.

The final class of feature selection algorithms is embedded approaches where the features are chosen while building the classifier. To represent this class two tree-based methods were adapted; random forest (RF) [87] and extra-trees (ET) [49]. The parameter considered was the number of trees used in each approach.

A summary of parameters of all considered feature selection methods along with the range of values searched for each parameter are given in Table 2.8.

Classification methods

An exhaustive comparison of all classification algorithms would be quite challenging. Therefore only a small number of approaches was explored, chosen to represent most common machine learning approaches used in bioinformatics. Identifying the cancer type from the NCI-60 dataset is inherently a multi-category classification problem. Therefore each considered approach must accommodate this setting or be adaptable by one-vs-one [51] or equivalent approaches. The types of algorithms tested fall into three main categories: linear, tree, and distance based methods. Again we used the Scikit-learn [81] Python implementations for all compared classification algorithms.

Linear classifiers use a linear function to score classes by taking the dot product of feature values and feature weights computed during training. One of the most powerful, flexible and ubiquitous linear classifier is the support vector machine (SVM) with linear kernel [3]. SVM has been utilized in numerous works describing predictive models with biological and medical significance. Both the penalty and loss function parameters were explored. Another powerful linear classifier is logistic regression (LR) [60]. The specific implementation uses one-vs-all to accommodate the multiclassification setting instead of the one-vs-one approach. The penalty function, and regularization parameters were explored.

Classification trees are a machine learning tool which has found extensive use in the biological and medical communities. This is partially due to both their resilience to over-fitting and ease of interpretation. This work looks at three related approaches; vanilla decision trees (DT) [5], random forest (RF) [49] and gradient boosting (GB) [106]. Decision trees represent class labels as leaves in the tree and branches are combinations of features that lead towards a leaf. Vanilla decision trees can often over-complicate the explanation necessary to arrive at the appropriate class label, however their interpretation is very simple. Random forest approach and gradient boosting are ensemble learning techniques where multiple trees are created and the final decision is some aggregate. These approaches are less-susceptible to over-fitting however they are often computationally intensive. The common parameter explored is the number of trees used and for gradient boosting the number of boosting stages.

Distance based methods surveyed are k-nearest neighbors (KNN), cosine (Cos) and correlation (Corr). Cosine and correlation are simple classifiers which calculates the distance to all training samples from the test sample and assigns the label based on the closest match. KNN is a slightly more advanced version of the same concept however only k neighbors are considered.

A summary of parameters of all considered classification algorithms along with the range of values searched for each parameter are given in Table 2.9.

Validation strategy

A common validation strategy used in evaluating machine-learning is k-fold cross-validation [39, 55]. Here the data is partitioned into k equal size subsets with each set used in turn for testing while the other $k-1$ subsets are used as training data. Care must be taken taken to avoid substantial biases [105] by ensuring feature selection is performed only on the data reserved for training. Since the approach presented here is also parameterizing for each distinct model, nested k-fold cross-validation is used to tune parameter values. This requires an additional cross-validation experiment on each training dataset, where a grid-search over the considered parameter range is performed. The inner phase identifies the best parameter values which are then used exclusively in the outer cross-validation. In order to build stronger evidence for the models performance, the outer cross-validation phase was repeated 100 times, however the parametrization was only performed in the first iteration. Biases towards selecting more complex models with more parameters or overly fine grid-steps are still a possibility, however nested cross-validation should largely mitigate them. More advanced techniques presented in [18] could be utilized in future iterations. Classification methods and embedded An outline of the validation strategy can be seen in Figure 2-1

The nested k-fold cross-validation strategy is computationally very intensive. With $4 \times 9 = 36$ models (combinations of feature selection and classifier) to evaluate, dozens of parameter values and different number of selected markers there can be upwards of 1,000,000 individual classifier runs per data type. The majority of the jobs occur in the inner cross-validation loop, and fortunately can all be run in parallel on a cluster or multi-core server. Also, a pre-filtering heuristic was applied to speed up the feature selection process. For all datasets with more than 1,000 features we retained only the top 1,000 features as ranked by the F-statistic prior to any additional feature selection.

To further validate the results on external datasets, eight primary tumor cohorts from The Cancer Genome Atlas (TCGA) were identified to match five NCI-60 tissue-of-origin cell lines; central nervous system, colon, non-small cell lung, ovarian, and renal. The mapping of the TCGA cohorts to the NCI-60 cell lines can be found in Table 2.10. The TCGA derived gene expression micro-array data was obtained from the Broad Institute's GDAC Firehose utility [7, 14, 9, 10, 11, 12, 6, 8, 13]. The presented pipeline was used to selection biomarkers, identify and train the most informative model using NCI-60 data [64]. Then its performance was tested using the TCGA derived data.

Metrics

There are numerous metrics used in evaluating the accuracy of a predictive model. One common metric is AUC, or *area under the receiver operating characteristic (ROC) curve*. The ROC curve is a plot of the true positive rate against the false positive rate. The AUC is then the area under this curve and is used as a single measurement of classifier performance. This definition is typically for binary classification tasks, however there are several extensions to multiclass classification problems [24]. Since the classes are equally represented in the NCI-60 dataset this work utilizes the multiclass metric, $AUC_{total} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i)$, where $AUC(c_i)$ is the typical binary classification AUC for class c_i and $p(c_i)$ is the prevalence in the data of class c_i.

ILP feature selection without replicates

The following ILP is a solution to the feature selection problem in absence of replicates. First an indicator variable $x_j \in \{0, 1\}$ is defined for all features. When set to 1 this variable indicates that its corresponding feature will be included in the solution set. Another indicator variable $y_{i_1,i_2} \in \{0, 1\}$ is defined for each $1 \leq i_1 \leq i_2 \leq n$. This indicates if a particular pair of classes is covered by a given feature.

The ILP is defined as follows:

$$\begin{array}{ll} \min & \sum_{j=1}^{p} x_j \\ \text{s.t.} & \sum_{j=1}^{p} x_j D_j(i_1, i_2) \geq y_{i_1,i_2} \quad \forall i_1, i_2 = 1, ..., n,\ i_1 < i_2 \\ & \sum_{1 \leq i_1 < i_2 \leq n} y_{i_1,i_2} = r \end{array}$$

The objective is to simply minimize the number of selected features. The first constraint ensures that when a feature is selected the pairwise coverage indicator variable is also selected. The second constraint ensures all classes are covered with some amount of *redundancy* when possible. It may occur that two classes are not distinguishable. The value of r can be calculated using a brute force search of all class pairs and all variables or by solving the maximum distinct points problem.

This ILP will be continually re-run until the number of chosen features equals θ as described in algorithm 1. After each iteration the chosen features are removed from D. It should be noted that given there are n cell types, then atleast $logn$ number of features are necessary to discriminate between all pairs. However typically n is small, and the number of available features is large enough that it is necessary to come up with a strategy to distribute the discriminating power. The above strategy will not only choose the minimum number of features but every class will have equal discriminating power.

After features are selected it is necessary to then create a predictive model to make classifications of unknown samples. This work utilizes the minimum distance classifier which assigns an unknown sample the label of the closest known sample according to some distance metric.

Algorithm 1 ILP Feature Selection

```
given $D$, distance matrix derived from $E$
$features = \{empty\}$
while $|features| \neq \theta$ do
    $r = calculate_r(D)$
    $feats = ILP(D)$
    $features+ = feats$
    $D = D - feats$
end while
```

2.3 Results and Discussion

This study is evaluating the effect of three parameters simultaneously: the model, the data type and the number of markers. Therefore conclusions about the best predictive model are presented from the perspective of each parameter individually. In Figure 2-2 an overview of the AUC for each model, in each data type at each number of markers is presented as a heatmap. The hotter entries represent higher AUC.

Model effects

The accuracy of the predictive models varies greatly, with the various combinations of feature selection and classification algorithms. If the feature selection and classification algorithms are grouped by class, a high-level ranking becomes much clearer. In Figure 2-3 the relative ranking of each model is indicated by color for each data type at each number of features. The RFE-Linear combination which uses SVM-RFE for feature selection and logistic regression or SVM for classification is the best performing model in almost all instances. Close behind is Ensembl-Linear, where in Table 2.4 it is clear that it performs only slightly worse than RFE-Linear.

If the data type and number of features are fixed the effects of the models can be explored further. As seen in Figure 2-4 the mRNA and protein data types consistently afford the best classification accuracy at both high and low number of markers. Although classifiers have relatively poor performance on SNP data for 8 markers, as the number of selected biomarkers increases best performing classifiers based on SNP data match or slightly outperform those based on gene and protein expression. The accuracy of all models is generally highest at a high number of markers. Therefore mRNA and SNP at 16 (Figure 2-5) and 64 (Figure 2-6) markers were chosen to demonstrate model effects. Surprisingly, the effect of classifier choice is small as seen in Figure 2-3. The models are grouped by feature selection algorithm. For RFE there is very little difference between all the classifiers except decision trees and gradient boosting which are consistently poor performers. The major differences appear between feature selection groups, where SVM-RFE is the best, random forest and extra trees have equivalent performance, and Anova is the worst.

This conclusion is contrary to that of [39], where it was found that the t-test univariate filter (of which Anova is considered a multiclass generalization) often performed the best for feature selection. This could be due to the differences in the underlying complexity of the question; namely in [39] the goal was to predict metastatic relapse, which is a binary question, using gene expression micro-arrays. In addition, no parameter tuning using nested CV or similar approach was performed in [39].

Although this study cannot prove that a particular feature selection or classification algorithm is best in a certain scenario, it does indicate that a thorough model selection step is advised.

The relatively small effect of classifier choice is interesting and unexpected. This indicates that much more care should be given to choosing the right features, as this has the biggest effect on model performance.

Data type effects

The rich selection of data types available for these cell lines provides the opportunity to compare the ability of many types of biological data to classify the tissue of origin of a tumor cell line. Some of these data types fundamentally reflect gene expression levels: mRNA, protein and microRNA. While the other two: CNV and SNP, are generally assumed to reflect genomic changes at large (CNV) and small (SNP) scales. Comparisons of data type effects at all marker sizes are best seen in Figure 2-4.

The transition from normal tissue to cancerous tissue is generally associated with changes at the level of both gene expression and the genome. Frequent mutations, genomic rearrangements and large scale changes in gene expression are all characteristic of oncogenic transformation. However, cancer cells also retain many, if not most, of the essential hallmarks of the tissue of origin of the cancer. In this study, we use the tissue of origin as the *ground truth* and measure the ability of each data type to correctly infer the tissue of origin of a sample based upon each data type.

A priori, we expect some of these data types to be better at this task than others. For instance, mRNA profiles are highly distinct between different tissue types. For this reason, even after oncogenic transformation, an mRNA transcriptional profile characteristic of the tissue of origin is expected to resemble that of the normal tissue, more than it would the transcriptional profile of tumors derived from other tissues. For this reason, we expect (and find) that mRNA transcriptional profiles reliably and accurately infer the tissue of origin of tumor cell lines. Similarly, protein expression profiles are also very reliable indicators of the tissue of origin of a tumor. microRNA profiles are less powerful than either mRNA or protein expression profiles, but still fairly powerful indicators of tissue of origin. The relative weakness of microRNA profiles compared to mRNA and protein expression profiles may in part result from lower tissue specificity of microRNA expression relative to mRNA and protein.

The ability of genomic data to infer the tissue of origin of the tumor is subject to a very different set of biological constraints than expression data. While expression data is expected to be approximately identical across tissues regardless of patient identity, and thus similar between tumors derived from the same tissue but from different individuals; genomic data is identical across normal tissues in an individual, and differs between individuals. Thus, at first glance, genomic data would be expected to track with the individual, and be a very poor predictor of the tissue of origin of a cancer. However, dramatic genomic alterations are a hallmark of cancer progression, and distinct genomic alterations are often found in distinct cancer types. Accordingly, we find that copy number variation is about as powerful as microRNA profiles at inferring the tissue of origin of a cancer cell. This is likely due to the preferential occurrence of specific DNA rearrangements in cancers derived from specific cell types [88]. The SNP arrays however, which measure the presence of specific alleles in a sample, show unexpectedly strong ability to infer the tissue of origin of these cancer cell lines. Indeed, their performance is similar to that of the mRNA and protein expression profiles (perhaps even better at high numbers of markers). This was unexpected as SNPs should be roughly identical across all tissues in an individual, and by and large, reflect an individual's ancestry. However, this phenomenon has been previously observed in the NCI-60 data, and was found to result from the fact that intensity of signal on the SNP array was actually reflecting SNP copy number at duplicated loci, and thus indirectly measuring likely gene expression levels, rather than homogenization of genotypic diversity [28]. This effect was strongest for linked SNPs, and appears to be the result of local gene copy number amplification, which in turn enables increased gene expression. Thus, the ability of SNP arrays to accurately infer tissue of origin of cancer cell lines appears to result from increased gene expression driven by local duplication and increase in copy number. As the CGH arrays used to profile the NCI-60 lines provide much lower genomic resolution than the SNP arrays, they are less powerful at detecting and exploiting this effect. This unexpected behavior of the SNP arrays used to characterize the NCI-60 lines could be addressed by utilizing newer SNP arrays that control for copy number such as the Affymetrix SNP6 platform.

Number of marker effects

As one uses more biomarkers to classify samples, one expects increased performance, the possibility of over fitting, and the appearance of a plateau beyond which additional markers do not increase the power of classification. However, the rate at which these changes occur as more markers are used to classify a sample can be very different for various types of data.

Our analysis shows that mRNA, protein, and SNP data all plateau at about the same AUC (∼0.97). However, each of these data types reaches the plateau at a different number of markers: mRNA plateaus between 16 and 32 markers, while protein plateaus at around 32 markers, and SNP does not reach the same AUC until 64 markers are used. This may be result from the fact that each of these markers appear to measuring aspects of gene expression, with decreasing directness (SNP) or coverage (protein), and thus power of discrimination. The mRNA arrays used to characterize the NCI-60 cell lines provide direct assessment of the activity of thousands of protein-coding genes, while the protein arrays measure only somewhat more than 300 proteins. With thousands of potential markers to choose from, the mRNA-based models can select informative markers from a larger marker pool, and thus maximize the performance of a gene expression-based model more quickly than the protein arrays, which are restricted to a small subset of the protein coding genes represented on the mRNA arrays. The more direct nature of the protein measurement (i.e. closer to the active biological component) does not appear to outweigh the disadvantage of the lower coverage in the starting set of protein markers. As discussed in the preceding section, the SNP array appears to be measuring, in part, gene expression levels resulting from the amplification of specific regions of the genome in specific cancer types. However, there is likely to be a complex and possibly heterogeneous and non-linear relationship between signal intensity on the SNP array, and gene expression levels. Thus, despite the very large number of markers to choose from on the SNP array, highly informative markers are not as abundant in this data as they appear to be in the mRNA data. As a result, many more SNP markers are required to achieve the same level of performance as mRNA-based markers. It is hard to predict how the power of SNPs to infer cancer type might change when newer arrays, that control for copy number changes, are used to characterize these cell lines.

Similarly, CNV and microRNA markers approach the same level of performance as one another, but do so at different rates. While microRNA markers plateau quickly (at about 16 markers) CNV markers require 64-96 markers to reach the same level of performance. The quick plateau of microRNA-based markers is likely due to the highly tissue-specific expression of a minority of microRNAs, and the more global expression of the remaining majority. Once the few highly informative microRNAs have been selected and used, adding more provides little additional classification power. In the case of CNVs, like SNPs these markers reflect changes in the cancer cells genome that can lead to changes in gene expression that are distinctive features of cancer subtypes. However, not only do the CNV markers suffer from the indirect relationship between the marker and gene expression expected for SNPs, they are also a much lower resolution marker than SNPs (megabases vs single bases), and far fewer CNVs were measured on the arrays, thus limiting the likelihood that the most informative CNVâĂŹs were available for selection. Thus, the power of the CNV biomarker panel climbs slowly.

Taken together, these observations suggest that the absolute performance of a given biomarker data type to classify a cancer can be understood in the context of: the number of available markers for the model to choose from, the power of the most informative markers in the set, and the directness with which the data type reflects an informative aspect of the sample biology. Data types with a large number of markers to choose from, that are closely related to the biology of the sample, are most likely to yield highly effective small biomarker panels. While data types with lower saturation (fewer markers measured), and/or a less direct relationship to the biological differences between samples, will require more markers to reach maximal performance.

Combined model, data type, and number of marker effects

Ultimately all parameters should be considered simultaneously when attempting to build the best targeted predictive model. In order to do this it is necessary to build a validation framework to explores all parameters fairly and efficiently. Although it is a difficult task it is not impractical and interesting nuances can be extracted.

In this study it was observed that at the lowest number of markers (8) mRNA and protein were the best data types for cancer identification. For mRNA, SNP and protein the SVM-RFE was the best feature selection choice and ET was the best classifier. For CNV and microRNA the best classifier was LR and ET respectively. Interestingly for all data types at 8 markers except CNV a tree based classifier performed the best as seen in Table 2.4. It is possible that if only a few biomarkers are considered the tree based approaches explicit enumeration of decisions may be better suited, however it should be noted that the linear classifiers are typically only marginally worse.

At the highest number of markers tests (96) both RFE and ET perform strongly on all data types, however LR is the best classifier for all types expect SNP where KNN is the best. Both of these classification tools are technically simple, yet they perform this best which lends credence to the Occam's razor principle which when applied to machine learning places preference on simpler explanations.

External validation

The amount of over-fitting when building a predictive model is always a concern. This effect was measured in an external validation experiment utilizing analogous gene expression micro-array data obtained from several studies which are part of the TCGA project [7, 14, 9, 10, 11, 12, 6, 8, 13]. The results of this comparison indicated that biomarker and model selection using AUC as the ranking criteria is robust and performs well across studies. In Table 2.5 it can be seen that colon (CO), CNS and renal (RE) cancer types were distinguishable with a high degree of accuracy using between 8 and 96 markers. The CNS type was more challenging to differentiate after 32 markers, while ovarian (OV) and lung cancers (LC) were extremely difficult to differentiate at any marker size.

The NCI-60 data is derived from decades old cell lines, while the TCGA was derived from recently sampled from primary solid tumor. Additionally the matched cancer types did not have comparable histological classification. Finally there three additional cancer types (ME, LE, BR) which were present in NCI-60 but not included in the external validation set. These classes were included in the training. Despite these differences the presented method was able to perform biomarker selection and build accurate predictive models for this challenging external validation experiment. A complete breakdown of the per-class prediction rate by cancer marker set size is provided in Table 2.11.

No replicates

Typically some type of k-fold cross-validation is used to evaluate the performance of a predictive model. However in this scenario leave-out-out cross-validation (LOO CV) on the n cell types is utilized. Assume that a lineage of the n cell types is available. At each iteration, a cell type is left out as the test cell type and the other $n1$ cell types are used to select a panel of q markers. This test cell type is then searched against the $n1$ cell types using the chosen marker panel.

Since a single cell type is left out, it cannot be matched against itself. The best case scenario is to map it to an adjacent cell type in the lineage. Hence, the score of a candidate is found by 2^{d1}, where d is the distance between the candidate and the true cell type in the given lineage. This way, a candidate cell type gets a score of 1 if it is adjacent to the true cell type. After n iterations, AUC can be calculated. To determine the size of a marker panel, a search of the number markers q in a range, e.g. $\{6, 7, , 96\}$.

	RFE/digi	RFE/expr	Greedy	ILP
SVM	97.84	97.91	97.78	97.87
Cosine	95.76	95.37	94.29	93.83
Correlation	96.97	96.84	96.42	96.23
Hamming	49.95	49.95	49.95	49.95

Table 2.3: AUC of evaluated methods

SCLD Data

In this experiment, we considered only markers in our curated data set of 28 cell types. Among these markers, 361 of them were found in the reference database. Therefore, the reference database can be viewed as a 28 x 361 expression matrix. Results can be seen in table 2.3.

2.4 Conclusion

The initial hypothesis motivating this research was that certain predictive models will perform better on different data types at different dimensionality. While this hypothesis holds, the difference in accuracy between models is often small and allows for several generalizations. Namely that RFE is clearly the best feature selection algorithm and both SVM and LR are the best classifiers as seen in Figures 2-2 and 2-3. Both mRNA and protein expression are the overall best performing data types for the cancer classification question. However to maximize predictive accuracy all models at all parameters should be parameterized and vetted fairly before conclusions are made.

The performance of the ILP on no-replicate data as seen in Table 2.3 indicates that it performs no worse, but no better than the recursive feature elimination approach given the right paired classifier. Interestingly there was little difference between the ILP and the greedy approach. However the flexibility of the ILP enables it to be easily adapted to other scenarios, such as the introduction of covariances, or class specific weights.

Table 2.4: Table of AUC for top performing models for each data type and grouped by marker set size.

#	SNP			mRNA			CNV			microRNA			Protein		
8	**RFE**	**ET**	**0.8598**	**RFE**	**ET**	**0.9585**	**RFE**	**LR**	**0.7198**	**RFE**	**RF**	**0.8352**	**RFE**	**ET**	**0.9426**
	RFE	RF	0.8591	RFE	RF	0.9554	ET	LR	0.7115	RFE	SVM	0.8352	ET	ET	0.9394
	RFE	SVM	0.8321	RFE	SVM	0.9521	RF	LR	0.71	RFE	KNN	0.8295	RFE	RF	0.9382
	ET	ET	0.8295	RFE	LR	0.951	RFE	ET	0.691	RFE	ET	0.8275	RF	ET	0.9376
				RFE	KNN	0.9467	RFE	RF	0.6802	Anova	SVM	0.8089	ET	RF	0.9312
										Anova	LR	0.8051	RF	RF	0.9272
										RF	ET	0.8028			
										RF	RF	0.8027			
										RFE	LR	0.8021			
										RF	LR	0.802			
16	**RFE**	**ET**	**0.922**	**RFE**	**ET**	**0.972**	**ET**	**LR**	**0.7616**	**RFE**	**SVM**	**0.8758**	**RFE**	**ET**	**0.9666**
	RFE	RF	0.9162	RFE	LR	0.9709	RFE	LR	0.7607	RFE	KNN	0.8704	ET	ET	0.9582
	RFE	SVM	0.9111	RFE	RF	0.9681	RF	LR	0.7468	RFE	RF	0.8671	RFE	RF	0.9565
	RFE	KNN	0.9033	RFE	SVM	0.968				RFE	ET	0.8597			
	ET	ET	0.8997	RFE	Cos	0.9663				RFE	LR	0.8535			
	RFE	LR	0.897							Anova	SVM	0.8496			
	RF	ET	0.896												
	ET	RF	0.8914												
32	**RFE**	**LR**	**0.9685**	**RFE**	**LR**	**0.9759**	**RFE**	**LR**	**0.8194**	**RFE**	**KNN**	**0.8806**	**RFE**	**ET**	**0.9792**
	RFE	SVM	0.9674	RFE	ET	0.9757				RFE	RF	0.8801			
	RFE	KNN	0.966	RF	LR	0.9747				RFE	ET	0.8717			
	RFE	ET	0.9646	RFE	Cos	0.9736				RFE	SVM	0.8679			
	RFE	RF	0.9577	RFE	RF	0.9734				RFE	LR	0.866			
				RFE	SVM	0.9734									
64	**RFE**	**KNN**	**0.9911**	**RF**	**LR**	**0.9789**	**RFE**	**LR**	**0.8379**	**RFE**	**KNN**	**0.8746**	**RFE**	**ET**	**0.979**
	RFE	LR	0.9892	RFE	LR	0.9777				RFE	LR	0.8688	RFE	LR	0.9782
	RF	LR	0.9862	RFE	Cos	0.977				RFE	RF	0.8682	RF	LR	0.9731
	RFE	SVM	0.9843	RFE	ET	0.976				RF	LR	0.8595	RFE	KNN	0.9727
	ET	LR	0.9837	RFE	RF	0.9757				RF	Corr	0.8585			
				RF	RF	0.9755				RFE	ET	0.8578			
				ET	LR	0.9741				RF	KNN	0.8574			
				RF	ET	0.9737				RFE	SVM	0.8568			
				RFE	SVM	0.9733				Anova	KNN	0.8564			
				ET	RF	0.9728				Anova	LR	0.8557			
				RFE	Corr	0.9709				ET	LR	0.8539			
										RFE	Corr	0.8537			
										ET	Corr	0.8536			

Continued on next page

– Continued from previous page

#	SNP			mRNA			CNV			microRNA			Protein		
										ET	KNN	0.852			
										RFE	Cos	0.8492			
96	**RFE**	**KNN**	**0.9933**	**RF**	**LR**	**0.9808**	**RFE**	**LR**	**0.847**	**RFE**	**LR**	**0.8697**	**RF**	**LR**	**0.979**
	RF	LR	0.9918	RFE	LR	0.9787	ET	LR	0.8292	RF	KNN	0.8657	RFE	LR	0.9779
	RFE	LR	0.9916	RF	RF	0.9774				RF	LR	0.8643	ET	LR	0.9768
	ET	LR	0.9909	RFE	Cos	0.977				ET	LR	0.8634	RFE	ET	0.9765
				RFE	RF	0.9762				RFE	RF	0.8633	ET	ET	0.9734
				ET	LR	0.9761				RF	Corr	0.863	RF	ET	0.973
				ET	RF	0.9758				ET	Corr	0.8629			
				RF	ET	0.9746				RFE	KNN	0.8628			
				RFE	ET	0.9744				ET	KNN	0.8613			
										Anova	KNN	0.8596			
										Anova	LR	0.8573			
										RFE	SVM	0.853			
										ET	RF	0.8483			
										RFE	Corr	0.8477			
										RF	SVM	0.8474			

Table 2.5: Accuracy of the top performing model for each cancer type and grouped by marker set size.

Marker Set Size	CO	OV	CNS	LC	RE
8	0.1673	0	1	0.3656	0.0138
16	0.9856	0.037	0.8246	0.686	0.7403
32	1	0.1111	0.9123	0.2384	0.8571
64	1	0.0741	0.5965	0.1163	0.8961
96	1	0.2593	0.5351	0.0116	1

Table 2.6: Data files and normalization strategy used for each data set.

Data Type	CellMiner Data Type	Norm.	CellMiner File Name
SNP	DNA: Affy 500K SNP	CRLMM	nci60_DNA__Affy_500K_SNP_CRLMM.txt.zip
mRNA	RNA: Affy HuEx 1.0	GCRMA	nci60_RNA__Affy_HuEx_1.0_GCRMA.txt.zip
CNV	DNA: aCGH Agilent 44K	AgelentFE	nci60_DNA__aCGH_Agilent_44K_AgilentFE.txt.zip
microRNA	RNA: microRNA OSU V3 chip	log2	nci60_RNA__microRNA_OSU_V3_chip_log2.txt.zip
Protein	Protein: Lysate Array	log2	nci60_Protein__Lysate_Array_log2.txt.zip

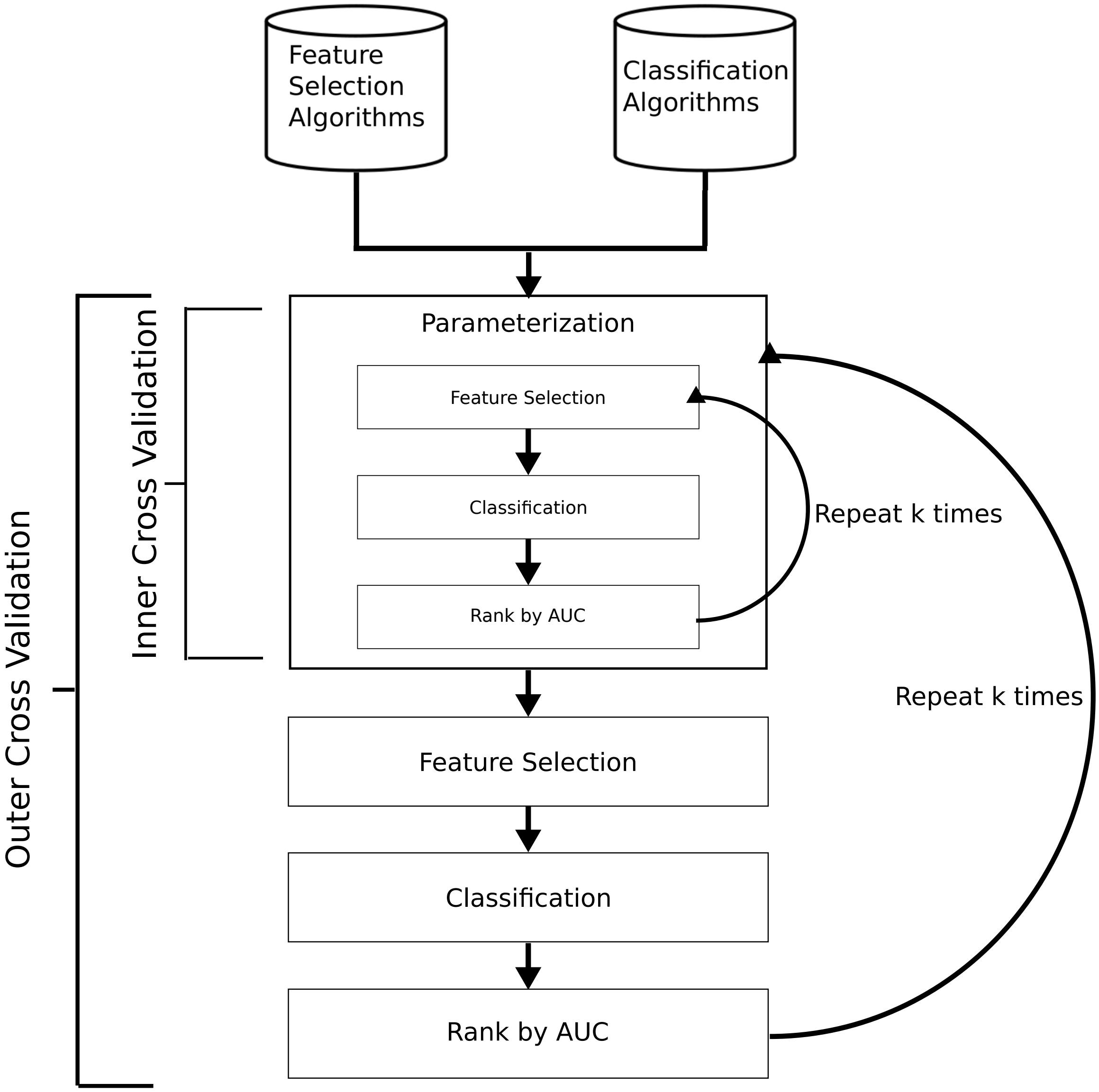


Figure 2-1: Flow chart of the validation strategy. First all combinations of feature selection and classification algorithms (4x9) are parameterized in the inner k-fold cross-validation loop based on the training folds of the outer k-fold cross-validation. The best parameters are found by maximizing AUC. Once the parameters are fix the outer k-fold cross-validation loop is run and the average AUC (or similar metric) is recorded.

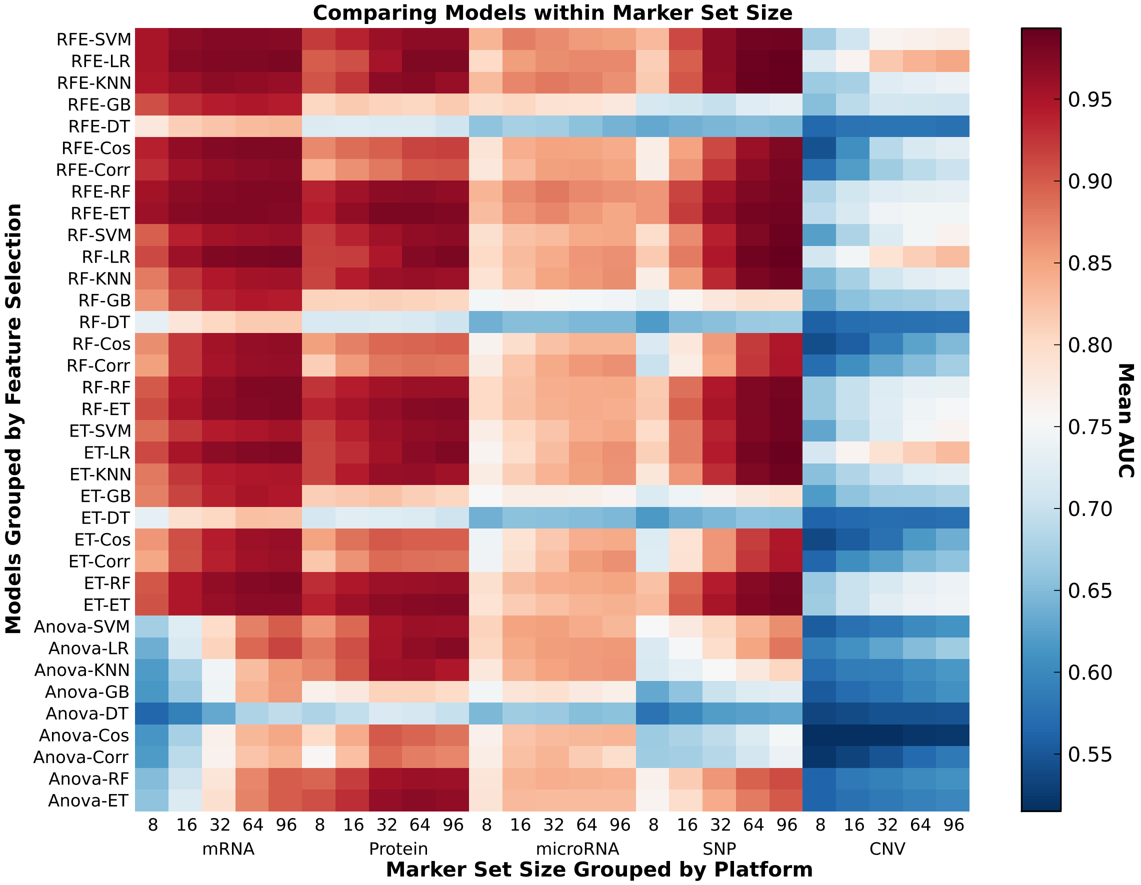


Figure 2-2: This heatmap contains the average AUC for each model (grouped by feature selection) for each data type at each number of markers. The darker the block, the more accurate the predictive model is.

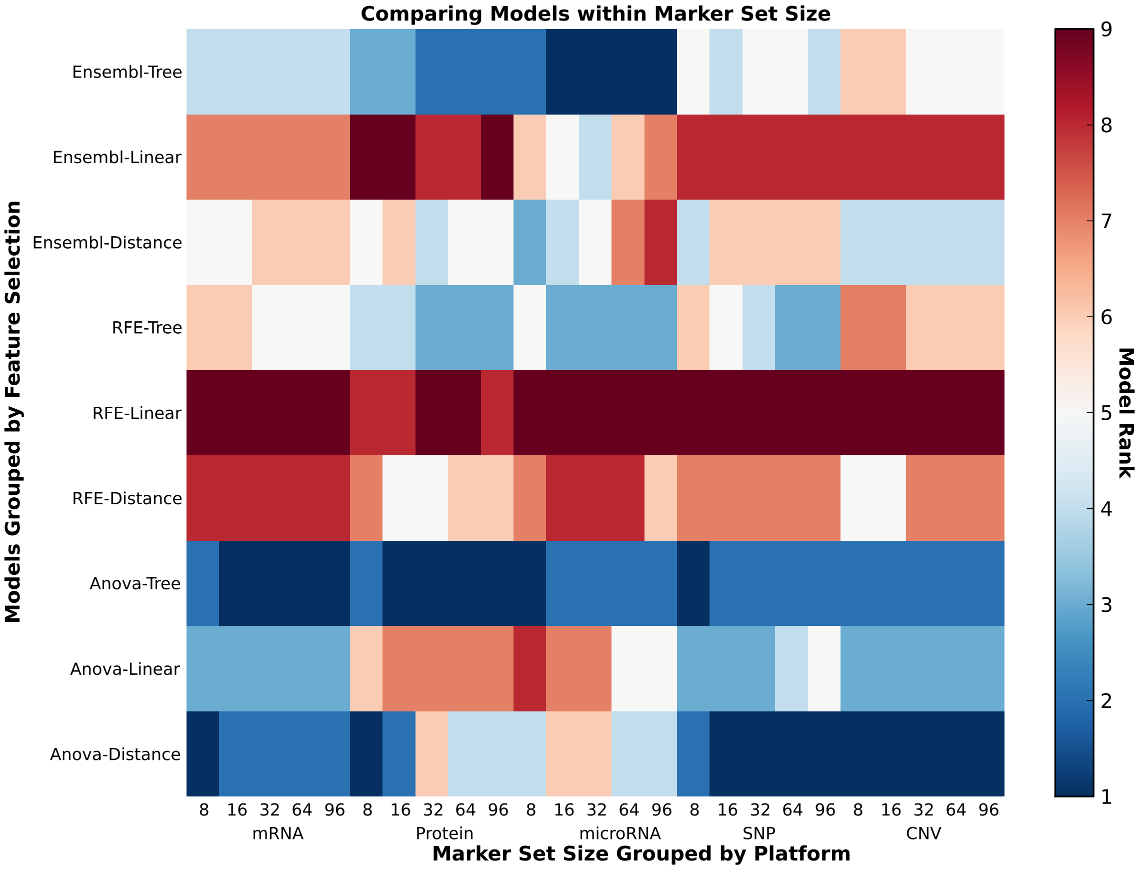


Figure 2-3: This heatmap contains the relative rank based on AUC of each model across all data types. The darker spots indicate higher AUC and rank.

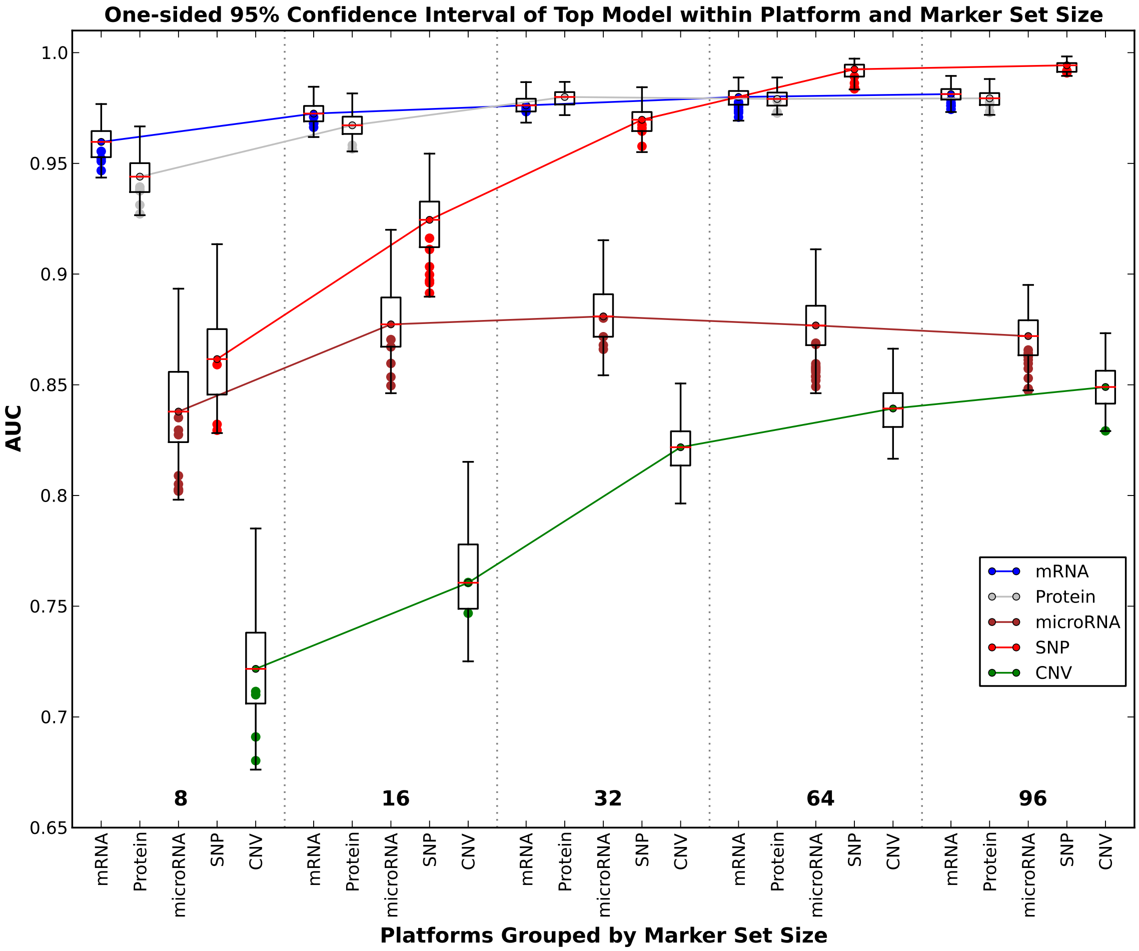


Figure 2-4: This figure contains box plots of the best model, for each data type and number of markers. The whiskers represent the 95% confidence interval, while the green dots represent another model with performance within the confidence interval.

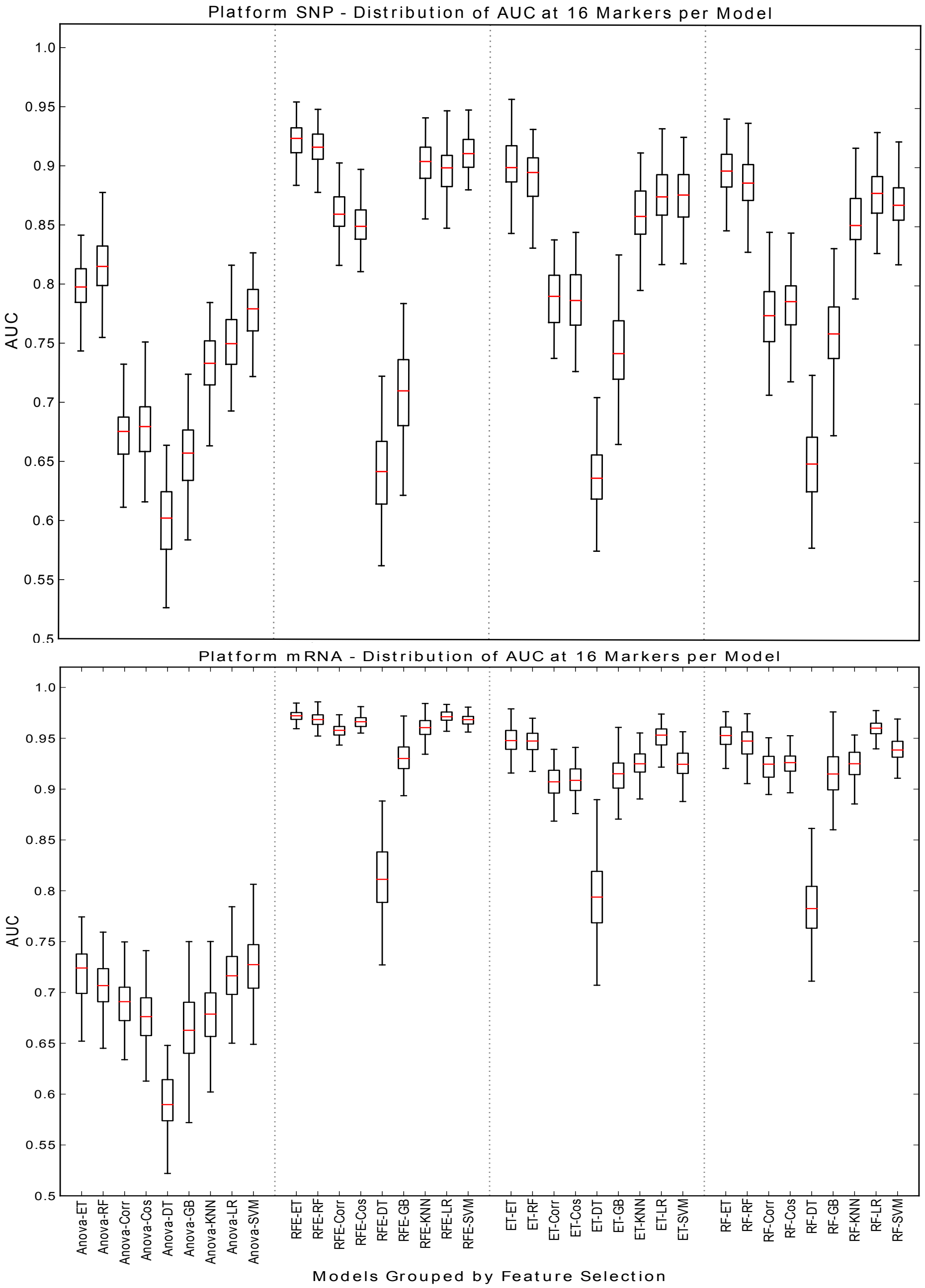


Figure 2-5: This figure contains box plots describing the AUC of each model, grouped by the feature selection component for SNP and mRNA data type at 16 markers.

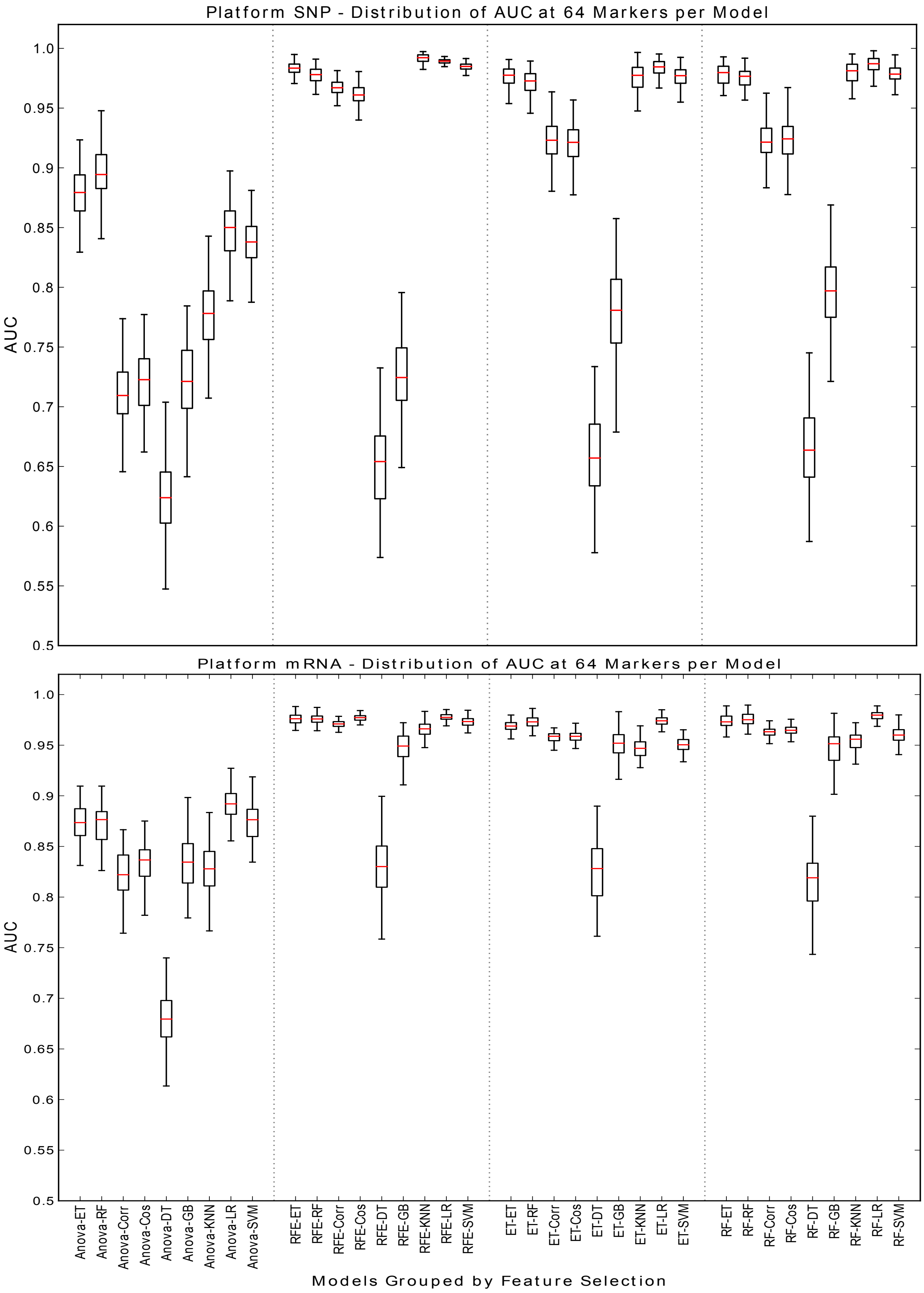


Figure 2-6: This figure contains box plots describing the AUC of each model, grouped by the feature selection component for SNP and mRNA data type at 64 markers.

Table 2.7: Cell Lines used in Analysis

Data	**Breast**	**CNS**	**Colon**	**Non-Small**	**Leukemia**	**Melanoma**	**Ovarian**	**Renal**
Type				**Cell Lung**				
mRNA	BR.MCF7	CNS.SF_268	CO.COLO205	LC.A549	LE.CCRF_CEM	ME.LOXIMVI	OV.IGROV1	RE.786_0
	BR.MDA_MB_231	CNS.SF_295	CO.HCC_2998	LC.EKVX	LE.HL_60	ME.MALME_3M	OV.OVCAR_3	RE.A498
	BR.HS578T	CNS.SNB_19	CO.HCT_116	LC.HOP_62	LE.K_562	ME.M14	OV.OVCAR_4	RE.ACHN
	BR.BT_549	CNS.SNB_75	CO.HCT_15	LC.HOP_92	LE.MOLT_4	ME.SK_MEL_2	OV.OVCAR_5	RE.CAKI_1
	BR.T47D	CNS.U251	CO.HT29	LC.NCI_H226	LE.RPMI_8226	ME.SK_MEL_28	OV.OVCAR_8	RE.RXF_393
			CO.KM12	LC.NCI_H23	LE.SR	ME.SK_MEL_5	OV.SK_OV_3	RE.SN12C
			CO.SW_620	LC.NCI_H322M		ME.UACC_257	OV.NCI_ADR_RES	RE.TK_10
				LC.NCI_H460		ME.UACC_62		RE.UO_31
				LC.NCI_H522		ME.MDA_MB_435		
						ME.MDA_N		
Protein	BR.MCF7	CNS.SF_268	CO.COLO205	LC.A549	LE.CCRF_CEM	ME.LOXIMVI	OV.IGROV1	RE.786_0
	BR.MDA_MB_231	CNS.SF_295	CO.HCC_2998	LC.EKVX	LE.HL_60	ME.MALME_3M	OV.OVCAR_3	RE.A498
	BR.HS578T	CNS.SF_539	CO.HCT_116	LC.HOP_62	LE.K_562	ME.M14	OV.OVCAR_4	RE.ACHN
	BR.BT_549	CNS.SNB_19	CO.HCT_15	LC.HOP_92	LE.MOLT_4	ME.SK_MEL_2	OV.OVCAR_5	RE.CAKI_1
	BR.T47D	CNS.SNB_75	CO.HT29	LC.NCI_H226	LE.RPMI_8226	ME.SK_MEL_28	OV.OVCAR_8	RE.RXF_393
		CNS.U251	CO.KM12	LC.NCI_H23	LE.SR	ME.SK_MEL_5	OV.SK_OV_3	RE.SN12C
			CO.SW_620	LC.NCI_H322M		ME.UACC_257	OV.NCI_ADR_RES	RE.TK_10
				LC.NCI_H460		ME.UACC_62		RE.UO_31
				LC.NCI_H522		ME.MDA_MB_435		
						ME.MDA_N		

Continued on next page

– Continued from previous page

Data Type	Breast	CNS	Colon	Non-Small Cell Lung	Leukemia	Melanoma	Ovarian	Renal
microRNA	BR.MCF7	CNS.SF_268	CO.COLO205	LC.A549	LE.CCRF_CEM	ME.LOXIMVI	OV.IGROV1	RE.786_0
	BR.MDA_MB_231	CNS.SF_295	CO.HCC_2998	LC.EKVX	LE.HL_60	ME.MALME_3M	OV.OVCAR_3	RE.A498
	BR.HS578T	CNS.SF_539	CO.HCT_116	LC.HOP_62	LE.K_562	ME.M14	OV.OVCAR_4	RE.ACHN
	BR.BT_549	CNS.SNB_19	CO.HCT_15	LC.HOP_92	LE.MOLT_4	ME.SK_MEL_2	OV.OVCAR_5	RE.CAKI_1
	BR.T47D	CNS.SNB_75	CO.HT29	LC.NCI_H226	LE.RPMI_8226	ME.SK_MEL_28	OV.OVCAR_8	RE.RXF_393
		CNS.U251	CO.KM12	LC.NCI_H23	LE.SR	ME.SK_MEL_5	OV.SK_OV_3	RE.SN12C
			CO.SW_620	LC.NCI_H322M		ME.UACC_257	OV.NCI_ADR_RES	RE.TK_10
				LC.NCI_H460		ME.UACC_62		RE.UO_31
				LC.NCI_H522		ME.MDA_MB_435		
						ME.MDA_N		
SNP	BR.MCF7	CNS.SF_268	CO.COLO205	LC.A549	LE.CCRF_CEM	ME.LOXIMVI	OV.IGROV1	RE.786_0
	BR.MDA_MB_231	CNS.SF_295	CO.HCC_2998	LC.EKVX	LE.HL_60	ME.MALME_3M	OV.OVCAR_3	RE.A498
	BR.HS578T	CNS.SF_539	CO.HCT_116	LC.HOP_62	LE.K_562	ME.M14	OV.OVCAR_4	RE.ACHN
	BR.BT_549	CNS.SNB_19	CO.HCT_15	LC.HOP_92	LE.MOLT_4	ME.SK_MEL_2	OV.OVCAR_5	RE.CAKI_1
	BR.T47D	CNS.SNB_75	CO.HT29	LC.NCI_H226	LE.RPMI_8226	ME.SK_MEL_28	OV.OVCAR_8	RE.RXF_393
		CNS.U251	CO.KM12	LC.NCI_H23	LE.SR	ME.SK_MEL_5	OV.SK_OV_3	RE.SN12C
			CO.SW_620	LC.NCI_H322M		ME.UACC_257	OV.NCI_ADR_RES	RE.TK_10
				LC.NCI_H460		ME.UACC_62		RE.UO_31
				LC.NCI_H522		ME.MDA_MB_435		
						ME.MDA_N		

Continued on next page

– Continued from previous page

Data	Breast	CNS	Colon	Non-Small	Leukemia	Melanoma	Ovarian	Renal
Type				Cell Lung				
CNV	BR.MCF7	CNS.SF_268	CO.COLO205	LC.A549	LE.CCRF_CEM	ME.LOXIMVI	OV.IGROV1	RE.786_0
	BR.MDA_MB_231	CNS.SF_295	CO.HCC_2998	LC.EKVX	LE.HL_60	ME.MALME_3M	OV.OVCAR_3	RE.A498
	BR.HS578T	CNS.SF_539	CO.HCT_116	LC.HOP_62	LE.K_562	ME.M14	OV.OVCAR_4	RE.ACHN
	BR.BT_549	CNS.SNB_19	CO.HCT_15	LC.HOP_92	LE.MOLT_4	ME.SK_MEL_2	OV.OVCAR_5	RE.SN12C
	BR.T47D	CNS.SNB_75	CO.HT29	LC.NCI_H226	LE.RPMI_8226	ME.SK_MEL_28	OV.OVCAR_8	RE.TK_10
		CNS.U251	CO.KM12	LC.NCI_H23	LE.SR	ME.SK_MEL_5	OV.SK_OV_3	RE.UO_31
			CO.SW_620	LC.NCI_H322M		ME.UACC_257	OV.NCI_ADR_RES	
				LC.NCI_H460		ME.UACC_62		
				LC.NCI_H522		ME.MDA_MB_435		
						ME.MDA_N		

Table 2.8: The tested parameters for each feature selection algorithm.

FS Method	Parameter	Description	Values
Anova	NA	No parameters	
RFE	Estimator	The supervised learning estimator	Support Vector Classification (SVC) with a linear kernal for the decision functions
	Estimator Parameter - C	The penalty parameter of the error term	0.25, 1, 4, 16, 64, 256
Random Forest	Max Features	Function to determine the number of features to consider when looking for best split	sqrt and log2
	N Estimators	The number of trees to be used in the forest	10, 50, 100, 250
Extra Trees	Max Features	Function to determine the number of features to consider when looking for best split	sqrt and log2
	N Estimators	The number of trees to be used in the forest	10, 50, 100, 250

Table 2.9: Classification Parameters

CL Method	Parameter	Description	Values
Support Vector Machine	Kernel	Function to use as a decision function	RBF and Linear
	RBF Gamma	Kernel coefficient for RBF	0.03125, 0.125, 0.5, 2, 8, 32, 128, and 512
	RBF C	Penalty Parameter of the error term	0.03125, 0.125, 0.5, 2, 8, 32, 128, and 512
	Linear C	Penalty Parameter of the error term	0.03125, 0.125, 0.5, 2, 8, 32, 128, and 512
Logistic Regression	Penalty	Norm used in the penalization	l2
	C	Inverse of regularization strength; smaller values specify stronger regularization	0.25, 1, 4, 16, 64, and 256
Decision Trees	Max Features	Function to determine the number of features to consider when looking for best split	sqrt and log2
Random Forest	Max Features	Function to determine the number of features to consider when looking for best split	sqrt and log2
	N Estimators	Number of trees to be used in the forest	10, 50, 100, 250
Extra Trees	Max Features	Function to determine the number of features to consider when looking for best split	sqrt and log2
	N Estimators	Number of trees to be used in the forest	10, 50, 100, 250
Gradient Boosting	N Boosting Stages	Number of boosting stages to perform	100, 250, and 500
	Max Depth	Maximum depth of the individual regression estimators	3, 5, 7
K-nearest Neighbors	Compute Nearest Neighbor	Algorithm used to compute the nearest neighbors	BallTree and KDtree
	Distance Function	Function used to calculate the distance	Euclidean and Manhattan

Continued on next page

– Continued from previous page

CL Method	Parameter	Description	Values
	Weight Function	A function to apply weights to the points	Uniform and Inverse weighting based on the distance to their neighbors; the closer the distance the better the score.
Cosine	NA	No parameters	
Correlation	NA	No parameters	

Table 2.10: External Validation: NCI-60 to TCGA mapping

NCI-60 Cell Line	TCGA Cell lines
Central Nervous System	Glioblastoma multiforme (GBM) Brain Lower Grade Glioma (LGG)
Lung	Lung adenocarcinoma (LUAD) Lung squamous cell carcinoma (LUSC)
Colon	Colon adenocarcinoma (COAD) Rectum adenocarcinoma (READ)
Ovarian	Serous Cystadenocarcinoma (OV)
Renal	Kidney renal clear cell carcinoma (KIRC) Kidney renal papillary cell carcinoma (KIRP)

Table 2.11: Accuracy per Cancer Type Grouped by Marker Set Size.

TCGA	**#**	**NCI-60 Cell Lines**							
Samples	**Markers**	**ME**	**LE**	**CO**	**CNS**	**RE**	**BR**	**OV**	**LC**
CNS	8	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
CO	8	0.0144	0.0048	0.1779	0.0769	0.0048	0.0192	0.4135	0.2885
LC	8	0.0291	0.0058	0.0058	0.3081	0.1977	0.0174	0.0	0.436
OV	8	0.4074	0.1852	0.0	0.0	0.0	0.0	0.0	0.4074
RE	8	0.0	0.0	0.0649	0.7013	0.0	0.0	0.1039	0.1299
CNS	16	0.0	0.0	0.0	0.8246	0.1754	0.0	0.0	0.0
CO	16	0.0	0.0	0.9856	0.0	0.0144	0.0	0.0	0.0
LC	16	0.0	0.0058	0.064	0.0523	0.1512	0.0407	0.0	0.686
OV	16	0.0	0.2222	0.037	0.0	0.0	0.0	0.037	0.7037
RE	16	0.013	0.026	0.0649	0.0519	0.7403	0.013	0.0649	0.026

Continued on next page

– Continued from previous page

TCGA	#	NCI-60 Cell Lines							
Samples	**Markers**	**ME**	**LE**	**CO**	**CNS**	**RE**	**BR**	**OV**	**LC**
CNS	32	0.0	0.0877	0.0	0.9123	0.0	0.0	0.0	0.0
CO	32	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
LC	32	0.0	0.0581	0.2151	0.1279	0.2791	0.0814	0.0	0.2384
OV	32	0.0	0.7037	0.1481	0.0	0.0	0.037	0.1111	0.0
RE	32	0.0	0.0779	0.0	0.013	0.8571	0.0	0.039	0.013
CNS	64	0.0	0.386	0.0	0.5965	0.0175	0.0	0.0	0.0
CO	64	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
LC	64	0.0	0.0872	0.064	0.0058	0.7035	0.0233	0.0	0.1163
OV	64	0.0	0.4074	0.1481	0.0	0.1111	0.0741	0.0741	0.1852
RE	64	0.013	0.0909	0.0	0.0	0.8961	0.0	0.0	0.0
CNS	96	0.0	0.4386	0.0	0.5351	0.0263	0.0	0.0	0.0
CO	96	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
LC	96	0.0	0.1802	0.2209	0.0174	0.5233	0.0465	0.0	0.0116
OV	96	0.0	0.4074	0.2222	0.0741	0.037	0.0	0.2593	0.0
RE	96	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0

Chapter 3

Single-cell Assisted Deconvolution

Separation of component signals from biological data has been an open challenge for a number of years. This problem is found in many other domains and is often referred to as signal separation problem. Recent advances in single cell genomics have enabled researchers to sample gene expression signatures of single cells and they are now able to accurately measure the canonical gene expression profiles of given cell types. Using this additional data type the following work will demonstrate how a simple existing deconvolution approach can be enhanced to yield greater accuracy than methods which do not take advantage of single cell data.

3.1 Motivation

Cell or tissue type heterogeneity is present in data collected from numerous biological sources. It is typically too difficult or impossible to physically separate cell types in any given mixture. Computational gene expression deconvolution is the process by which this separation is done in silico [68]. One application of deconvolution is assisting stem cell biologists in obtaining whole-transcriptome expression profiles of closely related cell types. Here we present an approach that utilizes single-cell qPCR probing of a small number of genes to aid in the deconvolution of whole-transcriptome profiles of mixed samples.

The expression profiles of m genes measured in n mixtures of k cell types are modeled as $X = SC$, where X is a $m \times n$ matrix whose columns are the expression profiles of individual *mixtures*, S is $m \times k$ *signature* matrix whose columns are expression profiles of individual cell types, and C is a $k \times n$ *concentration* matrix whose columns represent the proportions of each cell type in individual mixtures. There are a variety of approaches used to solve the deconvolution problem and they can be classified based on the expected input and output.

One formulation used by [102] propose the problem as estimating S when the mixtures X and concentrations C are known. Typically the concentrations are measured by some external technique such as FACS or FISH. Another formulation [33, 86, 2] is to assume X and S are known, and the goal is to estimate C. This formulation is often found when studying extensively studied cell types such as the blood lineage and is often referred to as supervised deconvolution. Unsupervised deconvolution is the simultaneous estimation of both S and C given X. This scenario is much more challenging, however several approaches have yielded promising results [97, 58, 112]. Finally the semi-supervised formulation assumes that existing information such as unique marker genes are available to describe each cell type [29].

This work proposes a novel variation on these previously proposed themes where in addition to the bulk mixture data, single-cell expression profiles denoted as $Z = z_1, z_2, ...z_q$ where q is the number of single cells available. Typically this comes from a microfluidic device that performs qPCR or RNA-Seq reactions on each single-cell. This additional data can be pre-processed into the canonical cell type signatures where typical supervised deconvolution approaches are applicable.

One common application of single cell genetics is the detection of rare cell types. These rare cells can be progenitor stem cells which will only ever be present at an extremely low abundance, or potentially the cell type is rare because it is a subtle response to changes in environmental conditions. The likelihood that these rare cells are sample in the single cell experiments is proportional to its frequency. This makes experimentally measuring extremely rare cells quite difficult. This work will present first steps towards computationally inferring the canonical signature of rare cell types which cannot be directly measured.

3.2 Problem Definition

The motivation of this work is to deconvolve population level (i.e. mixtures) given a small number of single-cell resolution measurements from the same sample. We will denote by m the number of genes, by k the number of cell types, and by n the number of samples.

The gene expression level measured by qPCR for a gene i is typically given as the threshold cycle C_T^i. This measurement varies logarithmically with the abundance level.

The C_T^i values are often normalized using a constantly expressed housekeeping gene (or the geometric average of several housekeeping genes) in order to account for variances in starting material which would effect detection threshold. Thus the normalized values are reported as

$$\Delta C_T^i = C_{Ttarget}^i - C_{Treferencegene}^i$$

The ΔC_T^i values can be converted to a linear scale by using

$$2^{-\Delta C_T^i}$$

as expression level.

A mixture is a heterogeneous collection of cells where the expression levels of m selected genes are measured using qPCR. A single measurement is denoted by the vector x, where $|x| = m$. A set of n mixtures is denoted as the mixture matrix X with dimensions $m \times n$.

The signature for a given cell type is denoted as a vector s, where $|s| = m$, and each element in the vector is the mean expression value of each gene in cells of this type. The complete set of signatures for all cell types is denoted as S where S is an $m \times k$ matrix.

Each mixture is assumed to be a linear combination of cell-type signatures. For one mixture the concentration of each cell type is denoted by a vector c, where

$|c| = k$. A set of n concentration vectors is denoted as the concentration matrix C with dimensions $k \times n$.

Thus, expression data of a set of mixtures is modeled as

$$X = SC$$

where each heterogeneous mixture is a linear combination of cell type signatures with concentrations specified by the columns of C.

3.3 Method

We use a two step approach whereby single cells are first clustered into representative cell types and the signature matrix is computed. Next the mixing proportions matrix C is estimated using quadratic programming.

3.3.1 Signature generation

One of the core challenges of single cell genetics is the meaningful classification or grouping of cells. For the purposes of this work the term cell type will be broadly defined as a cell phenotype that is statistically separable based on gene expression data. The signature matrix $S_{m\times k}$ is built by clustering the single-cell qPCR data $z_1, z_2, ...z_q$ into k clusters. This problem is an instance of unsupervised learning, where samples need to be labeled based on their gene expression vectors. Numerous objectives have been proposed such as minimizing the distance between samples in a cluster, and others focus on grouping functionally related samples. K-means clustering was chosen to group the single-cell data because it explicitly allows us to control the number of theoretical cell-types. The average expression profile of each single-cell in a cluster is used to create the cell-type signature matrix $S_{m\times k}$.

3.3.2 Concentration matrix

The next task is to solve for the concentration matrix $C_{k\times n}$. This is done utilizing the same methodology described in [33]. Each column of X corresponds to genes measured by qPCR, and is a linear combination of single cell expression profiles with unknown concentrations. Let us denote a particular column in X as x and its corresponding column in C by c. Inferring c can be formulated as the following quadratic program:

$$\begin{aligned} \text{minimize} \qquad & ||Sc - x||_2 \\ \text{subject to} \qquad & \sum c = 1 \\ & c_i \geq 0 \ \forall i = 0...m \end{aligned}$$

This least-squares formulation can be solved with any constrained quadratic programming solver.

3.3.3 Missing cell type

Solving for a missing cell type is an extension of the above computational deconvolution framework. The signature matrix including the missing cell type will be denoted as $\hat{S}_{m\times(k+1)}$, and the concentration matrix will be denoted $\hat{C}_{(k+1)\times n}$. The goal is to simultaneously calculate the missing kth column of signature matrix and the complete concentration matrix. The quadratic formulation is the following:

$$\begin{aligned} \text{minimize} \qquad & ||\hat{S}\hat{C} - X||_2 \\ \text{subject to} \qquad & \sum c = 1 \\ & c_i \geq 0 \ \forall i = 0...m \\ & s_k \geq 0 \end{aligned}$$

This is a non-linear non-negative least squares optimization problem. While problems of this nature can be quite challenging there is quite a large body of work dedicated towards efficiently solving these types of optimization problems. The technique known as damped least-squares (DLS) [56, 71] is used here. The missing cell type signature is initialized as the average of all the known cell types and the concentrations are set uniformly.

One additional challenge is determining if there is a missing cell type present in the mixtures. This is addressed by comparing the residual value of the system with and without the missing cell type, $\hat{r} = |X - \hat{S}\hat{C}|$, respectively $r = |X - SC|$. If $\hat{r} > r + \gamma$ where γ is some small constant then our approach will report that a missing cell type is likely present. The small constant is in place in order to require a non-trivial difference between the two scenarios.

3.3.4 Simulation

This work relies on two types of data, single cell and bulk data from the same source measured with the same technology. The combination of these data types is not available in any public data sources. Therefore it is necessary to simulate such datasets.

The single cell data used as the basis for the simulations comes from a publication which assessed different methods for measuring single cell gene expression [34]. Here 101 cells from 5 cell types (hematopoietic, intestinal, mammary gland, prostate and neural stem cells) were obtained and measured using the Fluidigm C1 and Biomark HD platforms. The 280 genes were chosen using a literature guided approach for known stem cell and lineage specific marker genes. The qPCR data was processed using Fluidigm software but no normalization to housekeeping genes was performed.

In order to properly asses the accuracy of the deconvolution and missing cell-type prediction a semi-continuous model for simulating both single cell and mixture data sets has been adopted. The main motivation behind this model is the bi-modal nature of gene expression observed in single cell experiments [73, 98, 34]. This previously noted property finds that genes are both absent and highly expressed in similar cells. The presented model therefore has two components; the probability of a gene being expressed in a cell, along with the the expression value of that gene when it is expressed. The presented model most closely mirrors the semi-continuous model presented in [73].

Active Transcription

Non-zero C_T^i value measurements of a gene i for a particular cell type l are assumed to follow a normal distribution, $N(\mu, \sigma^2)$. In our experiments the parameters of this normal distribution have been estimated from the training single cell data. The random variable is denoted as q_l^i.

Expression Probability

The probability that a particular gene is expressed and detected can be estimated directly by counting the number of non-zero expression values for a given gene i and cell type l. The random variable is denoted as p_l^i.

Combined Model

These two components can be combined in a variety of ways to create a model for simulating single-cell observations. For a given gene i and cell type l:

1. $W_l^i = p_l^i \cdot q_l^i$. Under this model each gene has a certain probably of being expressed, and when it is, the expression level is taken from the normal distribution fitted from the training data available for that gene and that cell type.

2. $W_l^i = q_l^i$. This is a special case of the previous model under which genes are assumed to always be expressed.

3. $W_l^i = \dfrac{1 + p_l^i}{2} \cdot q_l^i$. In this scenario only the fraction of the expression probability stemming from technical error is represented.

This bi-modal property can be observed by viewing the gene expression distribution of a random gene taken from the starting dataset. The true and simulated distributions are seen side-by-side in figure 3-1.

The procedure for creating simulated single cell for cell type i is:

1. for all $j = 1..m$ estimate p_j^i.

2. for all $j = 1..m$ estimate the parameters of the normal distribution and sample to obtain q^i_j

3. for all $j = 1..m$ multiply p^i_j by q^i_j

Two different scenarios were explored for simulating the concentration matrix C. The first is simply a uniform concentration of all cells. This implies a concentration of $\frac{1}{5}$ for this study. The second type was sampling uniformly at random from 0 to 1 exclusive for every cell type, then normalizing by the sum. This ensures the concentration of each cell type in a mixture sums to 1 and there are both high and low concentration cell types.

The mixtures were created by repeatedly sampling single cells according to the proportions dictated by C. The per gene expression levels were added together to create a single mixture. In this simulation, no appropriate housekeeping gene was available for normalization, so the gene expression levels for the mixtures were simply divided by the total number of cells. Normalizing single cell qPCR data is still an open concern [65, 74].

An error term is used in the simulations, $e = 5, 10, 25, 50, 100, 200$ which becomes which acts as the noise parameter and it represents the number of single-cell's used to create a particular mixture. The fewer single cells used, the more different that mixture will be than the sum of cell type signatures.

Unless otherwise stated each experiment was run 10 times and the presented results are the average of the 10 replicates.

3.4 Results and Discussion

3.4.1 Existing tools

There are several published tools [102, 33, 86, 2, 97, 58] for solving both the supervised and unsupervised gene expression deconvolution problem. The formulation presented here sits between these two versions of the problem. The completely supervised approaches are not able to adequately deconvolve the systems studied here because

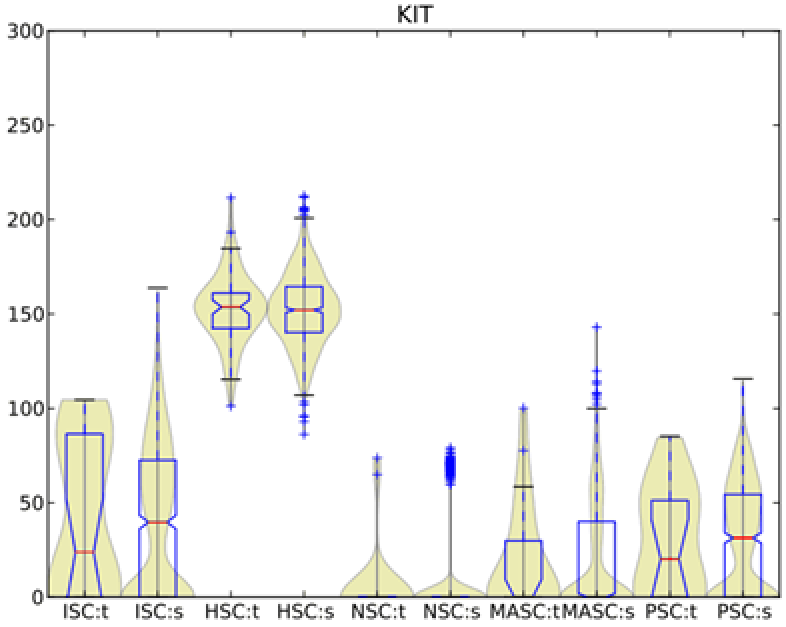


Figure 3-1: A violin plot of the true and simulated gene expression distribution of the KIT gene for each of the known cell types. The suffix s and t denotes the simulated and true distributions. The plot demonstrates that the simulated distributions closely resemble the true distributions.

the canonical cell type signature is required as input. Therefore only the unsupervised methods DSA [112], Deconf [86] and the semi-supervised NMF based methods SSKL, FROB [30, 29] are compared against our single cell quadratic programming based method denoted UCQP. The semi-supervised methods are given the opportunity to mine the single cell data to build the descriptive signatures necessary using tools built into the CellMix toolkit [30].

3.4.2 Experimental parameters

This work will compare the effect of several experimental parameters across the different deconvolution methods. One of the primary parameters explored is the number of mixtures measured in order to determine the effect of this parameter on accuracy. The error parameter is also varied to compare methods at varying degrees of difficulty, finally the number of genes used in the deconvolution is varied.

3.4.3 Mixture effects

One important dimension to this single cell aided deconvolution formulation is the number of mixtures necessary to properly deconvolve the system. Since both single cell and bulk qPCR data is necessary it is important that the number of mixtures required be kept small to contain costs. To this end several simulations were carried out where the number of mixtures is varied in the set $5, 10, 15, 20, 25, 30$.

Under the "ideal" scenario seen in figure 3-2a, the number of genes is fixed to 32, the error is set to 40 samples per cell, and the concentration is uniform across all mixtures. There is no absolute pattern or correlation between accuracy as the number of mixtures varies for any of the methods. The UCQP method outperforms both the semi and unsupervised methods.

In figure 3-2b the error and gene count is at 40 and 32 respectively, however the concentrations are random. There is again an absence of any connection between rmse as the number of mixture varies. The absolute rmse for DECONF, SSKL suffers slightly, UCQP remains the same and interestingly DSA, the unsupervised method, improves dramatically.

The final comparison in figure 3-2c fixes the error at 40 and the concentrations are uniform. Here the number of genes is halved from 32 to 16. The absolute rmse for all methods seems to suffer little, with a possible anti correlation between mixture count and accuracy observed for DECONF.

The number of mixtures used in deconvolution is a critical parameter as it influences the cost and feasibility of collecting the appropriate amount of data. Finding an algorithm which works well with a minimum number of mixtures necessary makes in-silico deconvolution more attractive. The UCQP approach presented here performs deconvolution one mixture at a time and therefore no constraints on the number of mixtures exist. It follows that the accuracy of UCQP should be independent of the number of mixtures. This was observed in 3-2. It should also be the case that DECONF, SSKL and DSA all have some dependency on the number of mixtures. However, only a minimal dependency is observed in figure 3-2.

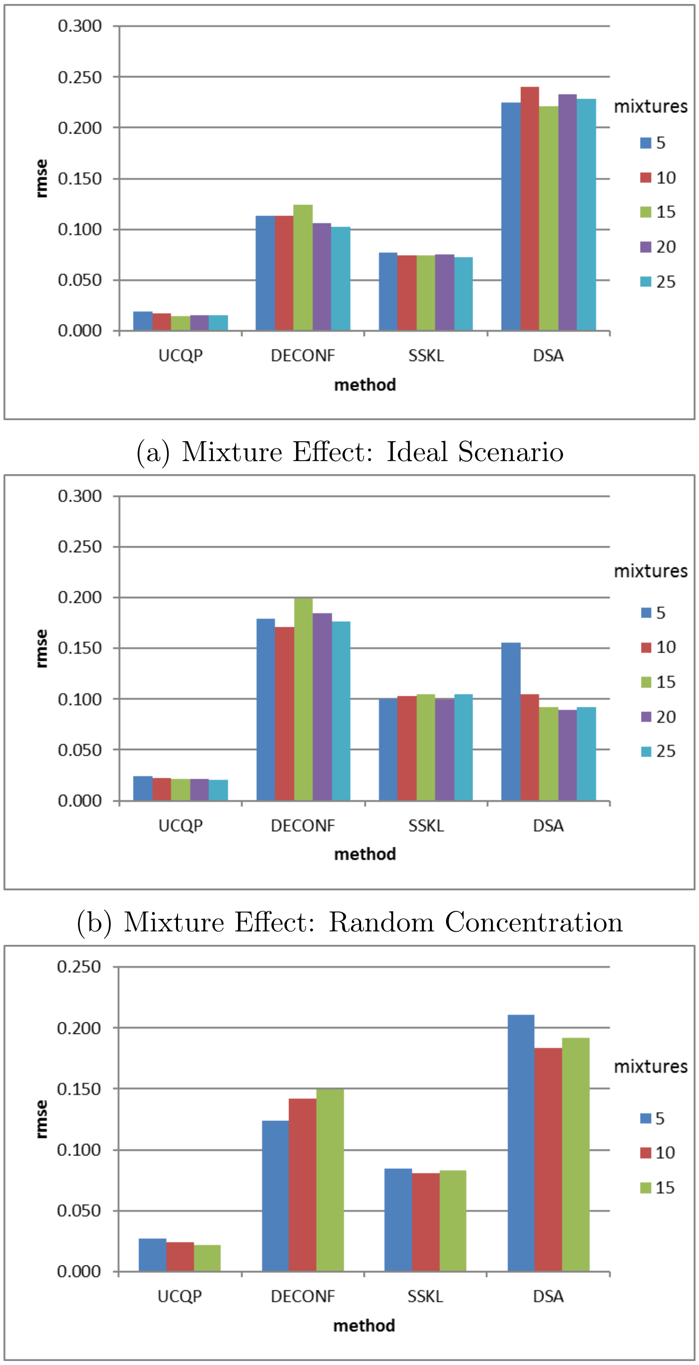
3.4.4 Error effects

It is important to quantify the ability of the deconvolution algorithm as the level of error changes. In the following simulation experiment the error parameter was varied between 5,20,40 and 100.

The ideal scenario has the number genes is fixed to 30, there are 10 mixtures and equal concentrations set across cell types. In figure 3-3a the accuracy of UCQP has a direct correlation with the error parameter. At 5 samples per mixture the algorithm performs worst and at 100 it has significantly lower error. The DECONF and DSA algorithms do not seem to have a clear relationship between error and accuracy and SSKL is not effected at all.

In figure 3-3b the number of mixtures is 10, gene count is 32 and the concentration is random. Both the UCQP and DSA algorithms have a direct correlation with the error parameter and rmse. The absolute rmse for UCQP is best for all scenarios.

In the last comparison in figure 3-3c the number of mixtures and gene count are 10 and 16, respectively, while the concentration is uniform. The results are almost



(a) Mixture Effect: Ideal Scenario

(b) Mixture Effect: Random Concentration

(c) Mixture Effect: 16 Genes

Figure 3-2: In this figure the accuracy of the 5 deconvolution methods in estimating the concentration matrix is compared using rmse metric. Each method is evaluated given a different number of mixtures between 5 and 25. In general no method has a strong correlation to the number of mixtures.

exactly the same as the ideal scenario in figure 3-3a.

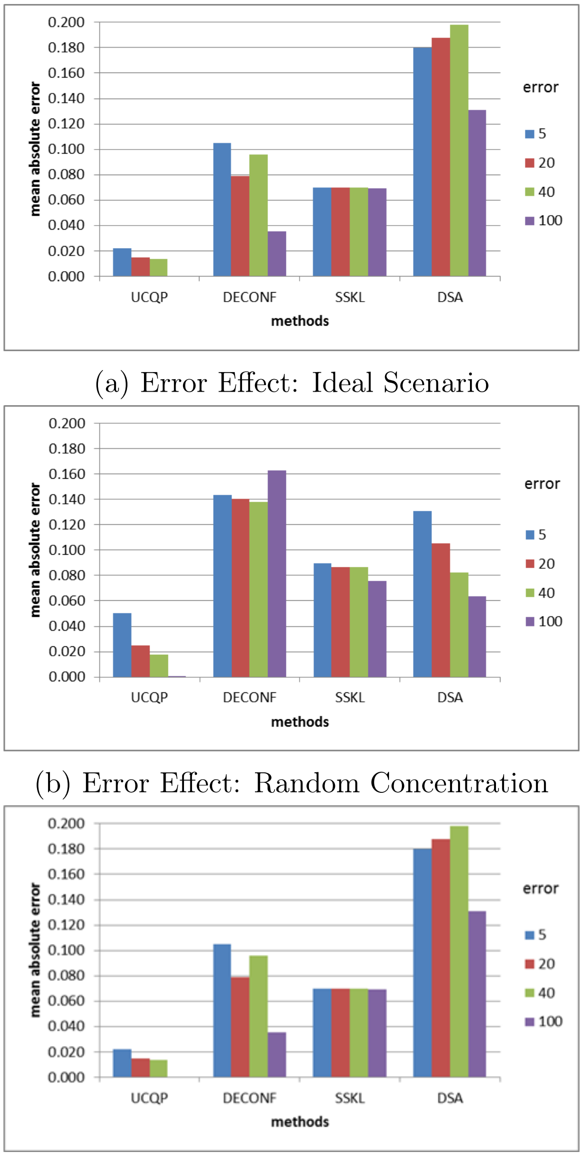
The proposed error model is quite simplistic and conflates both technical and biological noise. However without a stronger understanding of both the technical and biological sources a more advanced model may be premature. Adjusting the number of sampled single-cells to create each mixture enables the simulation to create easy scenarios, where each cell type is sampled sufficiently to get close to the canonical average, and more difficult instances where each cell type is far from the average. Additionally since this approach relies on sampling, it will inherently reflect the biological diversity of the supplied single-cells. If each cell population is quite diverse, more sampling will be required to find the average and therefore accurate deconvolution will be more difficult.

The error effects are most dramatic in 3-3b where the mixtures consist of random concentrations of each cell type. Intuitively this scenario seems the most challenging with respect to inferring cell type concentration and it seems natural that increased error makes the problem more challenging. Both UCQP and DCA improve as the sampling rate is increased, while the NMF methods do not respond as clearly. No explanation for these effect is immediately apparent. One advantage of the simpler quadratic programming based method is that understanding effects becomes easier.

3.4.5 Gene effects

The number of genes necessary to measure is an important parameter for this approach. Therefore a wide range of gene counts were surveyed in order to asses the effect across a variety of methods for qPCR. Values of 8, 16, 32, 64, 128 and 256 were tested. The genes were not selected at random, but rather were ranked using ANOVA test, excluding the lower scoring genes first.

In the ideal scenario depicted in figure 3-4a the error parameter is 40, concentrations are uniform and the number of mixtures is fixed at 5. Here UCQP outperforms all methods in terms of mean absolute error. There is a clear inverse relationship between the number of genes used and error in UCQP. No other method has this clear relationship.



(a) Error Effect: Ideal Scenario

(b) Error Effect: Random Concentration

(c) Error Effect: 16 Genes

Figure 3-3: In this figure the accuracy of the 5 deconvolution methods in estimating the concentration matrix is compared using rmse metric. Each method is evaluated given a different error level between mixtures between 5 and 100.

Figure 3-4b has the error fixed at 40, mixtures 5 and the concentrations are random. For UCQP this more challenging scenario has no noticeable effect on mean absolute error. There is still a clear inverse relationship between gene count and error for UCQP and no relationship for other methods.

The final scenario in figure 3-4c fixes the number of mixtures to 5, the concentrations are uniform but the error parameter is set to 5. Here the high error rate clearly changes the absolute error of UCQP, however the inverse relationship between gene count and error rate still exists.

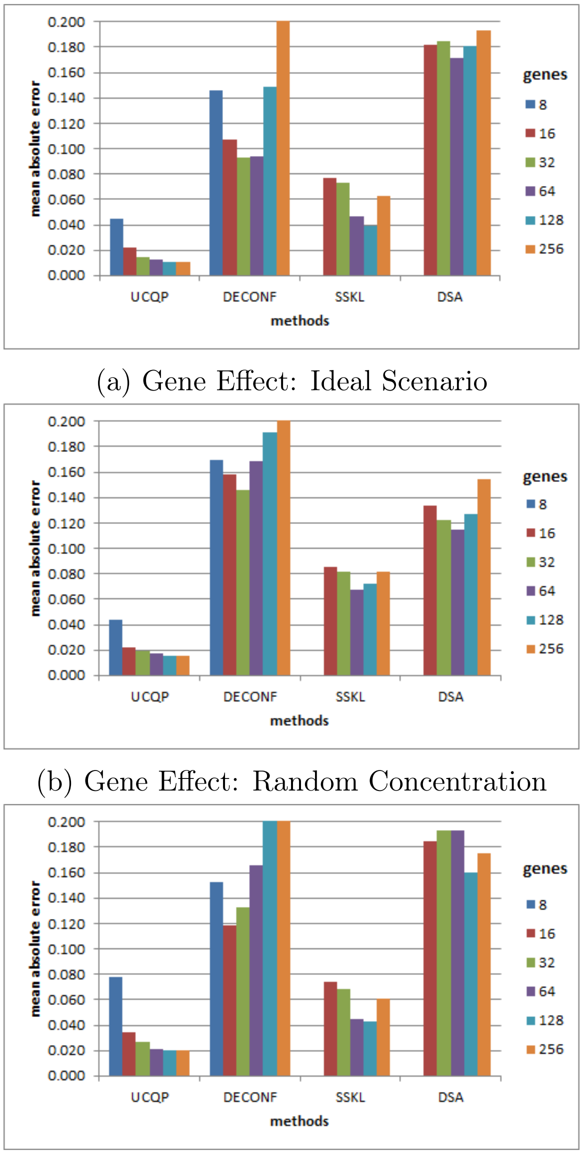
When considering just the UCQP method in figure 3-4 the effect of number of genes is quite clear. There exists a saturation point where the accuracy of the estimates no longer improves when adding more genes. This behavior is not observed in any of the other approaches. It should be noted that this simulation does not start with a random subset of genes, in fact all genes used in the data behind the simulation were chosen by domain experts. Also it is interesting to note that a high degree of accuracy can be obtained by using as few as 16 genes.

3.4.6 Missing Cell Type

Simultaneously estimating the canonical gene expression signature for a missing cell type and the concentrations of all cells is a difficult challenge. In order to test the proposed non-linear optimization approach a basic leave-one-out scenario was utilized. In this testing setup, each cell type was left out and its signature was estimated using the presented approach (UCQPM) compared against DECONF which can be told to estimate all cell types. As before, this scenario was repeated 10 times and the following results are the average of the 10 replicates. A second study our basic model used to detect if a cell type is missing or not. This was presented as a binary classification problem and assessed accordingly.

In general the UCQPM method outperforms unsupervised deconvolution on all of these missing cell type experiments. The results for these experiments are detailed in figures 3-5 and 3-6. This holds for all tested conditions.

In figure 3-5 the mixture effect is measured. There is a an inverse relationship



(a) Gene Effect: Ideal Scenario

(b) Gene Effect: Random Concentration

(c) Gene Effect: High Error

Figure 3-4: In this figure the accuracy of the 4 deconvolution methods in estimating the concentration matrix is compared using rmse metric. Each method is evaluated given a different gene count between 8 and 256.

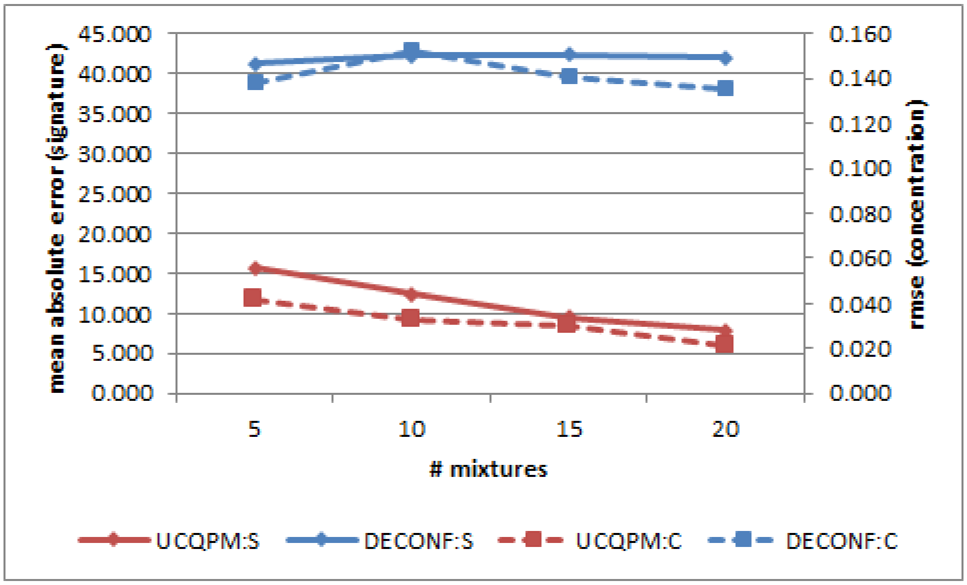


Figure 3-5: This graph has two y-axis, the left describes the mean absolute error of the signature estimate (solid line), while the right indicates the rmse of the concentration estimate (dotted line). The number of mixtures is fixed to 5, the error parameter is 100 and the concentrations are random. The x-axis varies the number of mixtures.

between both the concentration and signature estimates and error for UCQMP. This is consistent with results observed in the non-missing scenario. For DECONF no such consistent relationship is observed.

Figure 3-6 details the effect of noise on missing cell type estimates. The number of mixtures is fixed to 20, the concentration is random, and 64 genes are used. For UCQPM there is a clear inverse relationship between error and accuracy. At a high error rate (5 cells per type), both the signatures and concentrations are estimated poorly. As the error parameter increases the mean absolute error and rmse decreases which is also reflected in the non-missing scenario.

The final test evaluates our basic predictor of a missing cell type. The algorithm applies both missing and non-missing approaches to a given deconvolution problem, then applies the basic test $\hat{r} > r + \gamma$ where γ is some small constant. Where $\hat{r}$ is the residual calculated assuming a missing cell type, r is without a missing cell type. This scenario was tested by creating 10 mixtures with a missing cell type and 10 without. Both UCQPM and UCQP were run on each dataset and the test was applied predicting if it contained a missing cell type. This whole scenario was repeated

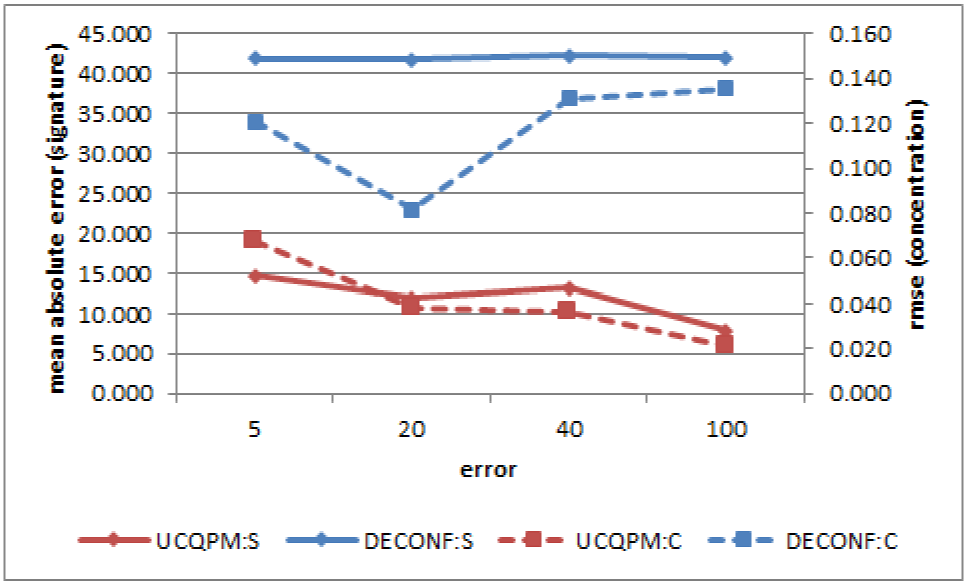


Figure 3-6: This graph has two y-axis, the left describes the mean absolute error of the signature estimate (solid line), while the right indicates the rmse of the concentration estimate (dotted line). The number of mixtures is fixed to 20, the gene count is 64, concentrations are random. The x-axis varies the error parameter.

10 times with the average results displayed in table 3.1.

error	accuracy	AUC
70	0.77	0.89
80	0.72	0.86
90	0.72	0.84
100	0.77	0.81

Table 3.1: This table records the binary classification problem attempting to predict the presence of a missing cell type from single cell and bulk qPCR data. There error column indicates the error parameter used in the simulation (higher being lower noise). The accuracy column is the typical notion of accuracy, and AUC is the area-under-curve.

In general this leave-one-out experiments indicates that UCQPM provides a more robust estimate of the missing cell type signature and concentrations than the unsupervised approach. This results is expected because the UCQPM utilizes more information about the system and is estimating a smaller number of variables . The quadratic programming formulation allows the method to be tailored to any number of unique situations, maximizing the use of available information.

However the ability of our simple threshold method to detect a missing cell type leaves much room for improvement. The best accuracy observed in this ideal experiment is 0.89 AUC as seen in table 3.1. Clearly this simplistic model will not be scalable to more complex scenarios where more than 1 cell type is missing. Alternative approaches will need to be explored.

3.5 Conclusions

This work presented an *in silico* gene expression deconvolution and missing cell type detection algorithm which is based on quadratic programming. This supervised approach relies on single cell resolution data to estimate cell type signatures, then uses a well defined non-negative least squares objective to estimate cell type concentrations. This approach was compared against leading semi-supervised and unsupervised NMF based methods and was shown to perform well. Estimating a missing cell type signature is also possible using a basic quadratic programming formulation. The signature and concentration estimation accuracy is much better than completely unsupervised NMF based methods. A basic approach to detecting a missing cell type was presented, however it is likely to be insufficient in more complex scenarios.

Chapter 4

Conclusions

The common theme in this work has been to find an elegant way of adopting computationally intensive optimization algorithms to solve problems unique to high-throughput genomics research. This final chapter summarizes the presented problems and strategies used to solve them. For each problem the current state of the art is discussed and potential future work proposed. Although only three problems and three strategies are discussed, the presented themes may aide others searching to find that edge necessary to solve their particular challenge.

4.1 Genome scaffolding

While genome assembly and scaffolding has always been a challenging task, this challenge has been exacerbated by the proliferation of sequencing technologies and the ensuing reduction in cost of whole genome sequencing. One part of a practical genome assembly is imparting order and orientation on contigs, a process known as scaffolding. The presented solution leverages the fixed-width nature of the scaffolding graph in conjunction with the ILP scaling technique NSDP in order to efficiently compute the best scaffolding with few compromises.

Assumptions and limitations

The scaffolding problem itself has only recently re-emerged as a relevant problem. As the human genome project neared completion scaffolding was formalized by Huson et. al. [47] in response the practical need of having to orient and organize the disparate contigs comprising the human genome. In this earlier context the technology available, Sanger sequencing, produced reads exceeding 1000 bases in length, forming large contigs. The paired reads were also fewer in number and due to their length aligned with less ambiguity. However the trend towards shorter contigs assembled from short reads has made the problem more difficult.

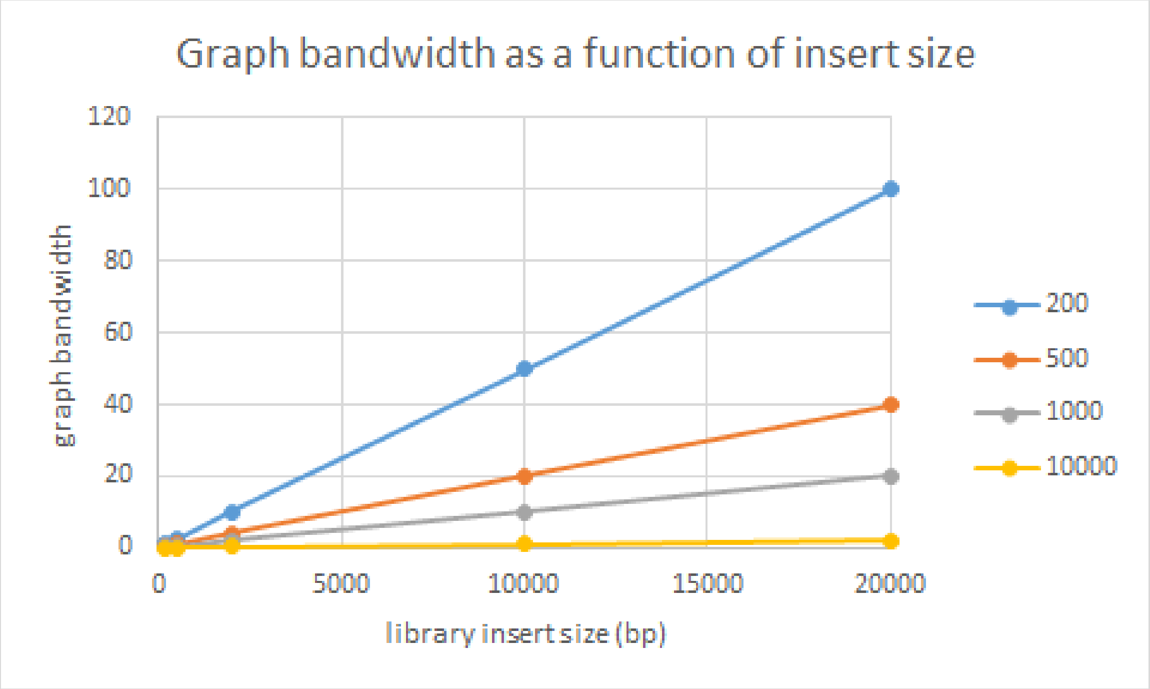


Figure 4-1: This plot indicates the expected upperbound on the scaffolding graph bandwidth for various minimum contig sizes and paired end library insert sizes. The x-axis indicates minimum contig size, up to 20k bases, and the y-axis is the graph bandwidth upperbound computed by $w \leq \frac{t}{l_{min}}$ where w is the graph bandwidth, t is the insert size and l_{min} is the minimum contig size. There are four series representing different minimum contig sizes.

Work in [26] made the observation that the scaffolding graph should have some special properties. Namely that the number of contigs which can be adjacent to any given contig is bounded by the maximum insert size of the paired end library. The cited formula $w \leq \frac{t}{l_{min}}$ from [95] indicates that in the ideal scenario the number of contigs is quite small. A graph representing a reasonable range of values can be see in Figure 4-1. This assumption, along with the presence of large *fence* [26] contigs bolsters the argument that the scaffolding graph can be decomposed into sufficiently small components solvable using NSDP.

While very promising, this assumption that the scaffolding graph has a fixed bandwidth fails in practice due to errors in the read alignment. The high repeat content of genomes causes increased assembly and read alignment errors. These errors significantly increase the number of possible adjacent contigs and thus make decomposition difficult.

State of the art and community work

The process of scaffolding fragments of a draft genome does not always rely solely on paired-end sequencing data. Recent work in [16, 91] explored the use of Hi-C or optical restriction maps to provide chromosome scale linkage information. These new techniques combine both paired-end sequencing containing short range information with the long range information to give a much more comprehensive scaffold. The combination of these two data types is often done in a hierarchical manner, first scaffolding on small or large scale then increasing or decreasing the resolution similar to [52].

Future work

Genome scaffolding will always be part of de novo genome assembly although the difficult of the problem may diminish as technology improves. Read lengths on NGS sequencers are again approaching several hundred base pairs, which improves both the contig assembly accuracy and the reliability of paired-read alignments. A recent publication [108] demonstrated the use of a basic k-mer counting algorithm for scaffolding using 1000 bp plus strobed reads from the Oxford Nanopore sequencer. Additionally the sheer volume of reads produced from traditional NGS sequencers, combined with more scalable and accurate genome assembly algorithms will again enable basic bundle size filtering [47].

An interesting application of de novo sequencing, and by extension scaffolding is the detection of structural variation (SV) [59]. The idea behind this work is that read alignment based approaches are inherently biased towards pre-programmed structural variations. Existing work performs complete de novo assembly, performs traditional SV detection, and uses an alignment based comparison of the assembled genome versus a reference for further evidence. While feasible, the cost of acquiring high coverage sequencing for de novo assembly still remains quite expensive. There may be an opportunity to utilize an ILP formulation similar to that used in SILP2 to detect certain structural variations. The high-level approach would be:

- align paired reads from subject as singletons against reference genome
- randomly fragment reference genome into pseudo-contigs
- construct scaffolding graph from paired end linkage information
- solve optimal orientation of pseudo-contigs

In this approach inversions would be detected when a pseudo-contig is flipped, since the contigs are known to originate from state A. Additional structural variation events could be detected by augmenting the ILP formulation.

4.2 Biomarker selection and predictive modeling

The process of biomarker selection is an extremely important step in translating genetic discovery to actionable medicine. It is also an active area of research, with many techniques being published. The work presented here provides an easy-to-use approach to survey all biomarker selection and classification algorithms to build the most accurate predictive model. The approach utilizes standard nested cross-validation techniques but has implemented them in a scalable cloud based architecture which relies on distributed task queues.

Assumptions and limitations

The main premise utilized in our pipeline is utilize extensive parametrization and large scale surveys to find optimal feature selection algorithm and build the best predictive models. Although somewhat simplistic in theory this strategy was proven to be quite effective in practice. However there are several areas where simplifying assumptions were made to make the strategy practical:

- normalization and scaling
- ignoring co-variance
- aggregate features

The notion of normalization and scaling is critical to building predictive models. Correction for technical and measurement noise can greatly improve performance and every algorithm is tuned to work in a specific numerical range. The presented work does not address normalization strategies which can be quite diverse, although the paradigm can be easily extended to accommodate them. Another simplification made was to ignore the co-variance between features when computing the optimal biomarker set. For some application, markers with strong co-variance may not be desirable. Finally recent advances in deep-learning and knowledge based feature selection have demonstrated the power of aggregate features, or feature sets. Here two or more genes are considered simultaneously or mathematically convolved into a single pseudo feature which can be more powerful.

State of the art and community work

Advances in predictive modeling and feature selection have been occurring at breakneck speed. Driving much of the cutting edge research are the fields of computer-vision and so called deep-learning. These techniques are making their way into the bioinformatics domain [110] with amazing results.

Future work

The modular nature of the parametrization and model selection pipeline enables easy integration of new approaches as they are published. Furthermore, the paradigm introduced in this work to solve the biomarker selection problem can be generalized to solve a variety of other problems. One such problem is dimension reduction and visualization of gene expression data.

There are numerous methodologies for clustering single cell gene expression profiles into cell types. One strategy is to enumerate each of the leading techniques and evaluate them using a variety of criteria as seen in Figure 4-2. This procedure enables the biologist to compare criteria and make an informed decision around the appropriate clustering procedure. This approach is an extension of existing ensemble and survey methodologies [22] made possible by a highly engineered cloud computing environment.

For dimension reduction we will test manifold learning methods (spectral embedding, t-SNE, linear embedding), PCA, sparsePCA, factor analysis and NMF. Several common clustering algorithms including affinity propagation, mean shift, spectral clustering, db scan and k-mean clustering as well as metrics such as silhouette score, gap statistic, Davies-Boulden, homogeneity index, separation index or some combination of these will also be tested. Often only intuition or experience is used to inform the method selection and parametrization. Rather than rely on intuition, our approach is to test each model against the others using several objective functions, allowing the most appropriate model for a particular dataset to be automatically selected. After global normalization of mean expression profiles, matching clustered single cell expression profiles to known cell type expression profiles will be done based on similarity measures selected via cross-validation experiments among both standard measures such as Pearson correlation and pathway-based similarity measures such as attract [69].

4.3 Single-cell assisted deconvolution

Single cell gene expression profiling is becoming widely accessible thanks to the proliferation of advanced sample preparation microfluidic equipment and protocols. Preliminary work has shown expression profiles at this resolution is quite different than bulk level gene expression. In order to bridge this gap we have proposed a novel usage of gene expression deconvolution using quadratic programming techniques. The effort presented here is still very much work in progress. The framework is still under development and requires extensive validation on biological datasets.

Assumptions and limitations

The primary assumption made in this work is that the researcher has the ability to cluster individual single cell expression profiles into meaningful clusters which accurately represent the constituent cell types. This assumption may necessitate sampling a very large number of single-cell data points. Cell-cycle effects may further result in a broad range of gene expression values for cells sampled from a single phenotypic cell type.

State of the art and community work

There have been many pioneering efforts to build solutions and techniques for dealing with the unique challenges found in single-cell analysis. For example a recent study explored techniques [15] for removing cell cycle effects when studying single-cells across multiple conditions. This can allow for better biomarkers selection and potentially yield better automated clustering.

Many groups are also investigating advanced dimension reduction and clustering [90, 101] which are more appropriate for the single-cell domain. Others are exploring lineage inference techniques [107, 70] where gene expression profiles are compared and ordered according to various objectives, or they are physically collected at different time points. The developmental lineage can then be computationally inferred using a variety of approaches.

Future work

While a universally agreed upon definition of what constitutes a cell type is still lacking, curating and organizing single cell, cell line, and bulk tissue expression data into a database of canonical cell type expression profiles can greatly simplify future gene expression studies. Establishing such a database would not only serve as a repository for single cell data, but, by accepting cell line and bulk mixtures (coupled with deconvolution), would allow it to become a unified source to cell type definitions. Below we describe a customizable and modular strategy for establishing such a database, called Cell Type DB. The database is composed of 3 distinct layers as illustrated in Figure 5: the raw expression layer, an organization layer, and finally an aggregation to cell type layer.

The raw gene expression layer accepts 3 types of data, single cell gene expression, cell line gene expression and finally bulk (heterogeneous mixtures) gene expression. Data which is pre-processed by an external party will be required to be normalized according to best practices and to account for batch effects. Datasets which are deposited directly, such as new bulk or single cell data can be pre-processed directly. This layer will likely be implemented using Apache HBase which provides excellent real-time /read/write access to data tables consisting of millions of columns (genes/isoforms) by billions of rows (gene expression profiles).

The organization layer's primary purpose is to stratify each individual gene expression profile into clusters corresponding to cell type, and then convert each cluster into a canonical expression profile. The procedure for this is different for each data type; single cell will be processed using the typical clustering methodologies, cell line data is highly homogeneous and require minimal preprocessing, while bulk data will be computationally deconvolved into its constituent cell types. This layer will be implemented in MongoDB, a NoSQL DB which excels at storing semi-structured data.

The final aggregation layer will consist of the computed cell type expression profiles. A query will consist of two components; the raw expression profiles to start with, the organization strategy. This module access layer enables the database to support multiple, well defined strategies for defining a cell type.

4.4 Closing

The cat-and-mouse game between bioinformatics and problem size driven by the state-of-the art biotechnology platforms will continue. Fortunately, as the cost of data acquisition drops, so too does the cost of computing resources. Although researchers will always be looking to improve the theoretical approaches to solving large scale optimization problems, there is still substantial opportunity to implement practical solutions. The common themes in these solutions will be leveraging powerful distributed computing resources and alternative problem formulations which better approximate the underlying biological systems.

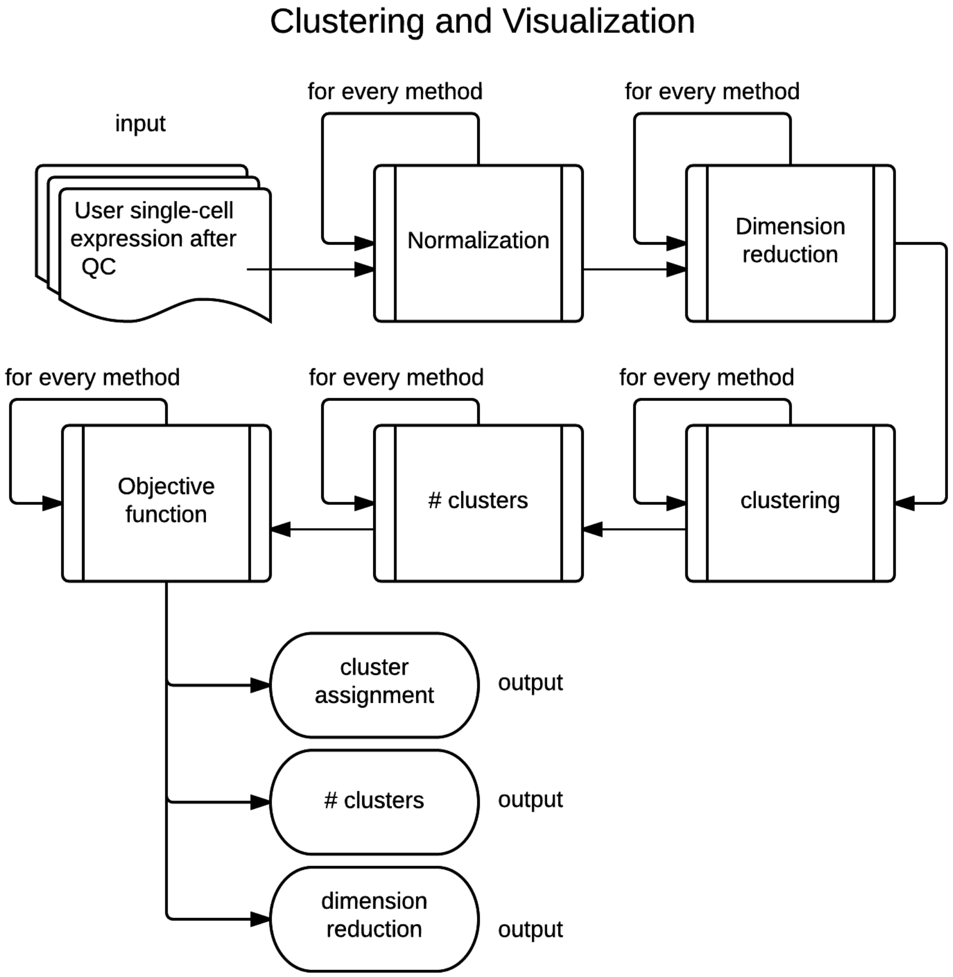


Figure 4-2: This algorithm tests all combinations of dimension reduction and clustering algorithms to find the approach which provides the optimal results.

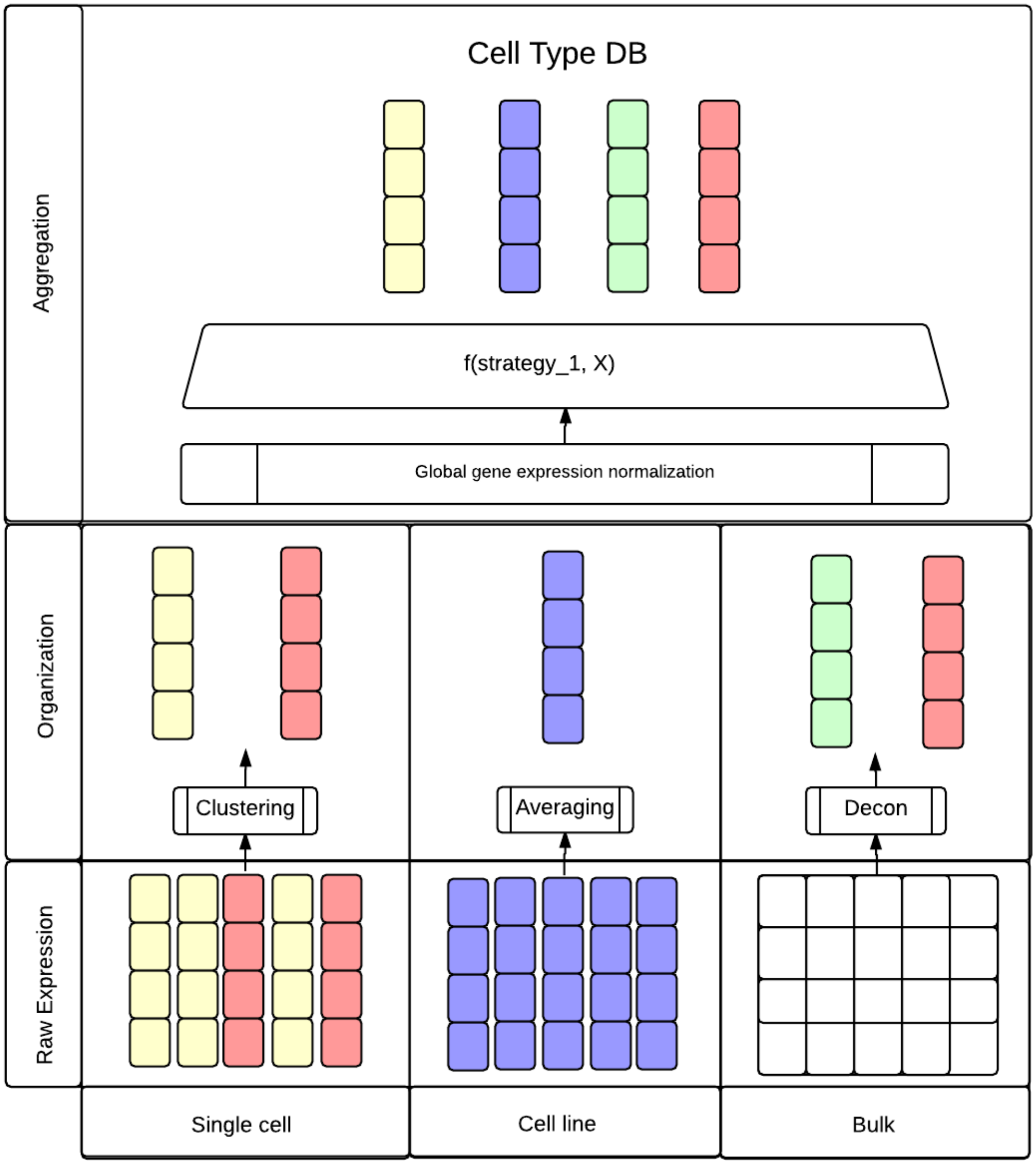


Figure 4-3: Cell Type DB provides a dynamic mechanism for storing and accessing canonical cell type expression profiles. Like a in a traditional database the first layer stores the raw single cell gene expression data. The middle layer is the cell annotation where one or more of the single cells are annotated according to the clustering methodologies or external mechanisms, the final layer applies a procedure to aggregate the single cell clusters into the canonical cell type expression profile.

Bibliography

[1] Thomas Abeel, Thibault Helleputte, Yves Van de Peer, Pierre Dupont, and Yvan Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, February 2010.

[2] Zeev Altboum, Yael Steuerman, Eyal David, Zohar Barnett-Itzhaki, Liran Valadarsky, Hadas Keren-Shaul, Tal Meningher, Ella Mendelson, Michal Mandelboim, Irit Gat-Viks, and Ido Amit. Digital cell quantification identifies global immune cell dynamics during influenza infection. *Molecular systems biology*, 10, 2014.

[3] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Scholkopf, and G. Ratsch. Support vector machines and kernels for computational biology. *PLoS computational biology*, 4(10):e1000173, Oct 2008. id: 1; LR: 20130605; JID: 101238922; RF: 73; OID: NLM: PMC2547983; 2008/10/31 [epublish]; ppublish.

[4] Keith R. Bradnam, Joseph N. Fass, Anton Alexandrov, Paul Baranay, Michael Bechner, Inanç Birol, Sébastien Boisvert, Jarrod A. Chapman, Guillaume Chapuis, Rayan Chikhi, Hamidreza Chitsaz, Wen-Chi C. Chou, Jacques Corbeil, Cristian Del Fabbro, T. Roderick Docking, Richard Durbin, Dent Earl, Scott Emrich, Pavel Fedotov, Nuno A. Fonseca, Ganeshkumar Ganapathy, Richard A. Gibbs, Sante Gnerre, Elénie Godzaridis, Steve Goldstein, Matthias Haimel, Giles Hall, David Haussler, Joseph B. Hiatt, Isaac Y. Ho, Jason Howard, Martin Hunt, Shaun D. Jackman, David B. Jaffe, Erich D. Jarvis, Huaiyang Jiang, Sergey Kazakov, Paul J. Kersey, Jacob O. Kitzman, James R. Knight, Sergey Koren, Tak-Wah W. Lam, Dominique Lavenier, François Laviolette, Yingrui Li, Zhenyu Li, Binghang Liu, Yue Liu, Ruibang Luo, Iain Maccallum, Matthew D. Macmanes, Nicolas Maillet, Sergey Melnikov, Delphine Naquin, Zemin Ning, Thomas D. Otto, Benedict Paten, Octávio S. Paulo, Adam M. Phillippy, Francisco Pina-Martins, Michael Place, Dariusz Przybylski, Xiang Qin, Carson Qu, Filipe J. Ribeiro, Stephen Richards, Daniel S. Rokhsar, J. Graham Ruby, Simone Scalabrin, Michael C. Schatz, David C. Schwartz, Alexey Sergushichev, Ted Sharpe, Timothy I. Shaw, Jay Shendure, Yujian Shi, Jared T. Simpson, Henry Song, Fedor Tsarev, Francesco Vezzi, Riccardo Vicedomini, Bruno M. Vieira, Jun Wang, Kim C. Worley, Shuangye Yin, Siu-Ming M. Yiu, Jianying Yuan, Guojie Zhang, Hao Zhang, Shiguo Zhou, and Ian F. Korf. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience*, 2(1):10+, July 2013.

[5] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. *Classification and Regression Trees*. Chapman & Hall/CRC, 1 edition, January 1984.

[6] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for brain lower grade glioma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[7] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for colon adenocarcinoma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[8] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for glioblastoma multiforme (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[9] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for kidney renal clear cell carcinoma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[10] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for kidney renal papillary cell carcinoma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[11] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for lung adenocarcinoma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[12] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for lung squamous cell carcinoma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[13] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for ovarian serous cystadenocarcinoma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[14] Broad. Broad institute tcga genome data analysis center (2014): Analysis overview for rectum adenocarcinoma (primary solid tumor cohort). Technical report, Broad Institute of MIT and Harvard, 2014.

[15] Florian Buettner, Kedar N. Natarajan, F. Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J. Theis, Sarah A. Teichmann, John C. Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33(2):155–160, January 2015.

[16] Joshua N. Burton, Andrew Adey, Rupali P. Patwardhan, Ruolan Qiu, Jacob O. Kitzman, and Jay Shendure. Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. *Nat Biotech*, 31(12):1119–1125, December 2013.

[17] J. Butler, I. MacCallum, M. Kleber, I. A. Shlyakhter, M. K. Belmonte, E. S. Lander, C. Nusbaum, and D. B. Jaffe. ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Research*, 18(5):810–820, February 2008.

[18] Gavin C. Cawley and Nicola L. C. Talbot. Preventing Over-Fitting during Model Selection via Bayesian Regularisation of the Hyper-Parameters. *J. Mach. Learn. Res.*, 8:841–861, May 2007.

[19] Mark J. Chaisson, Dumitru Brinza, and Pavel A. Pevzner. De novo fragment assembly with short mate-paired reads: Does the read length matter? *Genome research*, 19(2):336–346, February 2009.

[20] Markus Chimani, Carsten Gutwenger, Michael Jünger, Gunnar W. Klau, Karsten Klein, and Petra Mutzel. *The Open Graph Drawing Framework (OGDF)*, chapter 1, pages 77–90. CRC Press, 2012. to appear.

[21] Adel Dayarian, Todd Michael, and Anirvan Sengupta. SOPRA: Scaffolding algorithm for paired reads via statistical optimization. *BMC Bioinformatics*, 11(1):345+, 2010.

[22] A. Dobra. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, July 2004.

[23] Liat Ein-Dor, Or Zuk, and Eytan Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences*, 103(15):5923–5928, April 2006.

[24] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[25] Paul Flicek and Ewan Birney. Sense from sequence reads: methods for alignment and assembly. *Nature Methods*, 6(11s):S6–S12, 2009.

[26] Song Gao, Niranjan Nagarajan, and Wing-Kin Sung. Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. In *Proc. 15th Annual international conference on Research in computational molecular biology*, pages 437–451, 2011.

[27] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, STOC '74, pages 47–63, New York, NY, USA, 1974. ACM.

[28] L. A. Garraway, B. A. Weir, X. Zhao, H. Widlund, R. Beroukhim, A. Berger, D. Rimm, M. A. Rubin, D. E. Fisher, M. L. Meyerson, and W. R. Sellers. “Lineage addiction” in human cancer: lessons from integrated genomics. *Cold Spring Harbor symposia on quantitative biology*, 70:25–34, 2005.

[29] Renaud Gaujoux and Cathal Seoighe. Semi-supervised Nonnegative Matrix Factorization for gene expression deconvolution: a case study. *Infection, genetics and evolution : journal of molecular epidemiology and evolutionary genetics in infectious diseases*, 12(5):913–921, July 2012.

[30] Renaud Gaujoux and Cathal Seoighe. CellMix: a comprehensive toolbox for gene expression deconvolution. *Bioinformatics (Oxford, England)*, 29(17):2211–2212, September 2013.

[31] Arti Gaur, David A. Jewell, Yu Liang, Dana Ridzon, Jason H. Moore, Caifu Chen, Victor R. Ambros, and Mark A. Israel. Characterization of MicroRNA Expression Levels and Their Biological Correlates in Human Cancer Cell Lines. *Cancer Res*, 67(6):2456–2468, March 2007.

[32] Sante Gnerre, Iain MacCallum, Dariusz Przybylski, Filipe J. Ribeiro, Joshua N. Burton, Bruce J. Walker, Ted Sharpe, Giles Hall, Terrance P. Shea, Sean Sykes, Aaron M. Berlin, Daniel Aird, Maura Costello, Riza Daza, Louise Williams, Robert Nicol, Andreas Gnirke, Chad Nusbaum, Eric S. Lander, and David B. Jaffe. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences*, 108(4):1513–1518, January 2011.

[33] Ting Gong, Nicole Hartmann, Isaac S. Kohane, Volker Brinkmann, Frank Staedtler, Martin Letzkus, Sandrine Bongiovanni, and Joseph D. Szustakowski. Optimal Deconvolution of Transcriptional Profiling Data Using Quadratic Programming with Application to Complex Clinical Blood Samples. *PLoS ONE*, 6(11):e27156+, November 2011.

[34] Guoji Guo, Sidinh Luc, Eugenio Marco, Ta-Wei Lin, Cong Peng, Marc A. Kerenyi, Semir Beyaz, Woojin Kim, Jian Xu, Partha P. Das, Tobias Neff, Keyong Zou, Guo-Cheng Yuan, and Stuart H. Orkin. Mapping Cellular Hierarchy by Single-Cell Analysis of the Cell Surface Repertoire. *Cell Stem Cell*, 13(4):492–505, October 2013.

[35] Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, April 2013.

[36] Carsten Gutwenger and Petra Mutzel. A Linear Time Implementation of SPQR-Trees. In *Graph Drawing (GD'00)*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer-Verlag, 2001.

[37] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003.

[38] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46(1-3):389–422, 2002.

[39] Anne-Claire Haury, Pierre Gestraud, and Jean-Philippe Vert. The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures. *PLoS ONE*, 6(12):e28210+, December 2011.

[40] E. Hemphill, J. Lindsay, C. Lee, I.I. Mandoiu, and C.E. Nelson. Feature selection and classifier performance on diverse biological datasets. *BMC Bioinformatics*, 15(Suppl 13):S4, 2014.

[41] Edward E. Hemphill, Asav P. Dharia, Chih Lee, Caroline M. Jakuba, Jason D. Gibson, Frederick W. Kolling, and Craig E. Nelson. SCLD: a stem cell lineage database for the annotation of cell types and developmental lineages. *Nucleic Acids Research*, 39(suppl 1):D525–D533, January 2011.

[42] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158, 1973.

[43] Mark Howison, Felipe Zapata, and Casey W. Dunn. Toward a statistically explicit understanding of de novo sequence assembly. *Bioinformatics*, 29(23):2959–2963, 2013.

[44] Chi-Ming Huang, Yen-Chung Lin, Yenn-Jiang Lin, Shih-Lin Chang, Li-Wei Lo, Yu-Feng Hu, Chern-En Chiang, Kang-Ling Wang, and Shih-Ann Chen. Risk stratification and clinical outcomes in patients with acute pulmonary embolism. *Clinical Biochemistry*, June 2011.

[45] Martin Hunt, Chris Newbold, Matthew Berriman, and Thomas D Otto. A comprehensive evaluation of assembly scaffolding tools. *Genome biology*, 15(3):R42+, March 2014.

[46] Daniel H. Huson, Knut Reinert, and Eugene W. Myers. The greedy path-merging algorithm for contig scaffolding. *J. ACM*, 49(5):603–615, 2002.

[47] Daniel H. Huson, Knut Reinert, and Eugene W. Myers. The greedy path-merging algorithm for contig scaffolding. *J. ACM*, 49(5):603–615, September 2002.

[48] IBM. IBM ILOG CPLEX Optimizer.
urlhttp://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[49] Pierre G. June. Extremely Randomized Trees. In *Machine Learning*, volume 36, 2003.

[50] RichardM Karp. Reducibility Among Combinatorial Problems. In Michael Jünger, Thomas M. Liebling, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, editors, *50 Years of Integer Programming 1958-2008*, pages 219–241. Springer Berlin Heidelberg, 2010.

[51] Knerr. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In FrançoiseFogelman Soulié, editor, *Neurocomputing*. Springer Berlin Heidelberg, 1990.

[52] Sergey Koren, Todd J. Treangen, and Mihai Pop. Bambus 2: scaffolding metagenomes. *Bioinformatics*, 27(21):2964–2971, 2011.

[53] Carmen Lai, Marcel Reinders, Laura V. Veer, and Lodewyk Wessels. A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets. *BMC Bioinformatics*, 7(1), 2006.

[54] Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nat Meth*, 9(4):357–359, April 2012.

[55] Jae W. Lee, Jung B. Lee, Mira Park, and Seuck H. Song. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, 48(4):869–885, April 2005.

[56] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[57] Ruiqiang Li, Hongmei Zhu, Jue Ruan, Wubin Qian, Xiaodong Fang, Zhongbin Shi, Yingrui Li, Shengting Li, Gao Shan, Karsten Kristiansen, Songgang Li, Huanming Yang, Jian Wang, and Jun Wang. De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20(2):265–272, February 2010.

[58] Yi Li and Xiaohui Xie. A mixture model for expression deconvolution from RNA-seq in heterogeneous tissues. *BMC Bioinformatics*, 14(Suppl 5):S11+, 2013.

[59] Yingrui Li, Hancheng Zheng, Ruibang Luo, Honglong Wu, Hongmei Zhu, Ruiqiang Li, Hongzhi Cao, Boxin Wu, Shujia Huang, Haojing Shao, Hanzhou Ma, Fan Zhang, Shuijian Feng, Wei Zhang, Hongli Du, Geng Tian, Jingxiang Li, Xiuqing Zhang, Songgang Li, Lars Bolund, Karsten Kristiansen, Adam J. de Smith, Alexandra I. F. Blakemore, Lachlan J. M. Coin, Huanming Yang, Jian Wang, and Jun Wang. Structural variation in two human genomes mapped at single-nucleotide resolution by whole genome de novo assembly. *Nature Biotechnology*, 29(8):723–730, July 2011.

[60] J G G. Liao and Khew-Voon V. Chin. Logistic regression for disease classification using microarray data: model selection in a large p and small n case. *Bioinformatics*, May 2007.

[61] Yong Lin, Jian Li, Hui Shen, Lei Zhang, Christopher J. Papasian, and Hong-Wen Deng. Comparative Studies of de novo Assembly Tools for Next-generation Sequencing Technologies. *Bioinformatics*, 2011.

[62] J. Lindsay, H. Salooti, I.I. Mandoiu, and A. Zelikovsky. Ilp-based maximum likelihood genome scaffolding. *BMC Bioinformatics*, 15(Suppl 9):S9, 2014.

[63] J. Lindsay, H. Salooti, A. Zelikovsky, and I.I. Mandoiu. Scalable genome scaffolding using integer linear programming. In *Proc. BCB*, pages 377–383, 2012.

[64] Hongfang Liu, Petula D'Andrade, Stephanie Fulmer-Smentek, Philip Lorenzi, Kurt W. Kohn, John N. Weinstein, Yves Pommier, and William C. Reinhold. mrna and microrna expression profiles of the nci-60 integrated with drug activities. *Molecular Cancer Therapeutics*, 9(5):1080–1091, 2010.

[65] Kenneth J. Livak, Quin F. Wills, Alex J. Tipping, Krishnalekha Dattaa, Rowena Mittala, Andrew Goldson, Darren W. Sexton, and Chris C. Holmes. Methods for qPCR gene expression profiling applied to 1440 lymphoblastoid single cells. *Methods*, IN PRESS.

[66] C. Lo, A. Bashir, V. Bansal, and V. Bafna. Strobe sequence design for haplotype assembly. *BMC Bioinformatics*, 12 Suppl 1, 2011.

[67] P. L. Lorenzi, W. C. Reinhold, S. Varma, A. A. Hutchinson, Y. Pommier, S. J. Chanock, and J. N. Weinstein. DNA fingerprinting of the NCI-60 cell line panel. *Molecular cancer therapeutics*, 8(4):713–724, Apr 2009.

[68] Peng Lu, Aleksey Nakorchevskiy, and Edward M. Marcotte. Expression deconvolution: A reinterpretation of DNA microarray data reveals dynamic changes in cell populations. *Proceedings of the National Academy of Sciences*, 100(18):10370–10375, September 2003.

[69] Jessica C. Mar, Nicholas A. Matigian, John Quackenbush, and Christine A. Wells. attract: A method for identifying core pathways that define cellular phenotypes. *PloS one*, 6(10):e25445+, October 2011.

[70] Eugenio Marco, Robert L. Karp, Guoji Guo, Paul Robson, Adam H. Hart, Lorenzo Trippa, and Guo-Cheng Yuan. Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proceedings of the National Academy of Sciences*, 111(52):E5643–E5650, December 2014.

[71] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[72] Padma Maruvada and Sudhir Srivastava. Joint National Cancer Institute-Food and Drug Administration Workshop on Research Strategies, Study Designs, and Statistical Approaches to Biomarker Validation for Cancer Diagnosis and Detection. *Cancer Epidemiology Biomarkers & Prevention*, 15(6):1078–1082, June 2006.

[73] Andrew McDavid, Lucas Dennis, Patrick Danaher, Greg Finak, Michael Krouse, Alice Wang, Philippa Webster, Joseph Beechem, and Raphael Gottardo. Modeling Bi-modality Improves Characterization of Cell Cycle on Gene Expression in Single Cells. *PLoS Comput Biol*, 10(7):e1003696+, July 2014.

[74] Andrew McDavid, Greg Finak, Pratip K. Chattopadyay, Maria Dominguez, Laurie Lamoreaux, Steven S. Ma, Mario Roederer, and Raphael Gottardo. Data exploration, quality control and testing in single-cell qPCR-based gene expression experiments. *Bioinformatics (Oxford, England)*, 29(4):461–467, February 2013.

[75] Paul Medvedev and Michael Brudno. Maximum likelihood genome assembly. *Journal of Computational Biology*, 16(8):1101–1116, 2009.

[76] Jason R. Miller, Sergey Koren, and Granger Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327, 2010.

[77] Robert K Neely, Jochem Deen, and Johan Hofkens. Optical mapping of dna: Single-molecule-based methods for mapping genomes. *Biopolymers*, 95(5):298–311, 2011.

[78] C.H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[79] R. M. Parry, W. Jones, T. H. Stokes, J. H. Phan, R. A. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M. D. Wang. k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. *The Pharmacogenomics Journal*, 10(4):292–309, August 2010.

[80] Konrad H. Paszkiewicz and David J. Studholme. *De novo* assembly of short sequence reads. *Briefings in Bioinformatics*, 11(5):457–472, 2010.

[81] Fabian Pedregosa, Ga&#235;l Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and &#201;douard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2013.

[82] Mihai Pop. Genome assembly reborn: recent computational challenges. *Briefings in Bioinformatics*, 10(4):354–366, 2009.

[83] Mihai Pop, Daniel S. Kosack, and Steven L. Salzberg. Hierarchical scaffolding with Bambus. *Genome research*, 14(1):149–159, 2004.

[84] Atif Rahman and Lior Pachter. CGAL: computing genome assembly likelihoods. *Genome Biology*, 14(1):R8+, 2013.

[85] William C. Reinhold, Margot Sunshine, Hongfang Liu, Sudhir Varma, Kurt W. Kohn, Joel Morris, James Doroshow, and Yves Pommier. CellMiner: A Web-Based Suite of Genomic and Pharmacologic Tools to Explore Transcript and Drug Patterns in the NCI-60 Cell Line Set. *Cancer Research*, 72(14):3499–3511, July 2012.

[86] Dirk Repsilber, Sabine Kern, Anna Telaar, Gerhard Walzl, Gillian F. Black, Joachim Selbig, Shreemanta K. Parida, Stefan H. Kaufmann, and Marc Jacobsen. Biomarker discovery in heterogeneous tissue samples -taking the in-silico deconfounding approach. *BMC bioinformatics*, 11(1):27+, January 2010.

[87] Jeremy Rogers and Steve Gunn. Identifying Feature Relevance Using a Random Forest. In *Subspace, Latent Structure and Feature Selection*, pages 173–184. Springer-Verlag, 2006.

[88] A. V. Roschke, G. Tonon, K. S. Gehlhaus, N. McTyre, K. J. Bussey, S. Lababidi, D. A. Scudiero, J. N. Weinstein, and I. R. Kirsch. Karyotypic complexity of the NCI-60 drug-screening panel. *Cancer research*, 63(24):8634–8647, Dec 15 2003.

[89] R. S. Roy, K. C. Chen, A. M. Segupta, and A. Schliep. Sliq: Simple linear inequalities for efficient contig scaffolding. *arXiv:1111.1426v2[q-bio.GN]*, November 2011.

[90] Assieh Saadatpour, Guoji Guo, Stuart H. Orkin, and Guo-Cheng Yuan. Characterizing heterogeneity in leukemic cells using single-cell gene expression analysis. *Genome Biology*, 15(12):525+, December 2014.

[91] Subrata Saha and Sanguthevar Rajasekaran. Efficient and scalable scaffolding using optical restriction maps. *BMC Genomics*, 15(Suppl 5):S5+, July 2014.

[92] Leena Salmela, Veli Mäkinen, Niko Välimäki, Johannes Ylinen, and Esko Ukkonen. Fast scaffolding with small independent mixed integer programs. *Bioinformatics*, 27(23):3259–3265, December 2011.

[93] Steven L. Salzberg, Adam M. Phillippy, Aleksey Zimin, Daniela Puiu, Tanja Magoc, Sergey Koren, Todd J. Treangen, Michael C. Schatz, Arthur L. Delcher, Michael Roberts, Guillaume MarÃğais, Mihai Pop, and James A. Yorke. Gage: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research*, 22(3):557–567, 2012.

[94] Sevtap Savas, Laurent Briollais, Irada Ibrahim-zada, Hamdi Jarjanazi, Yun H. Choi, Mireia Musquera, Neil Fleshner, Vasundara Venkateswaran, and Hilmi Ozcelik. A Whole-Genome SNP Association Study of NCI60 Cell Line Panel Indicates a Role of Ca2+ Signaling in Selenium Resistance. *PLoS ONE*, 5(9):e12601+, September 2010.

[95] James B. Saxe. Dynamic-Programming Algorithms for Recognizing Small-Bandwidth Graphs in Polynomial Time. *SIAM Journal on Algebraic Discrete Methods*, 1(4):363–369, December 1980.

[96] Michael C. Schatz, Arthur L. Delcher, and Steven L. Salzberg. Assembly of large genomes using second-generation sequencing. *Genome Research*, 20(9):1165–1173, 2010.

[97] Russell Schwartz and Stanley E. Shackney. Applying unmixing to gene expression data for tumor phylogeny inference. *BMC bioinformatics*, 11(1):42+, January 2010.

[98] Alex K. Shalek, Rahul Satija, Xian Adiconis, Rona S. Gertner, Jellert T. Gaublomme, Raktima Raychowdhury, Schraga Schwartz, Nir Yosef, Christine Malboeuf, Diana Lu, John J. Trombetta, Dave Gennert, Andreas Gnirke, Alon Goren, Nir Hacohen, Joshua Z. Levin, Hongkun Park, and Aviv Regev. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453):236–240, May 2013.

[99] Uma T. Shankavaram, William C. Reinhold, Satoshi Nishizuka, Sylvia Major, Daisaku Morita, Krishna K. Chary, Mark A. Reimers, Uwe Scherf, Ari Kahn, Douglas Dolginow, Jeffrey Cossman, Eric P. Kaldjian, Dominic A. Scudiero, Emanuel Petricoin, Lance Liotta, Jae K. Lee, and John N. Weinstein. Transcript and protein expression profiles of the NCI-60 cancer cell panel: an integromic microarray study. *Molecular Cancer Therapeutics*, 6(3):820–832, March 2007.

[100] Oleg Shcherbina. Nonserial Dynamic Programming and Tree Decomposition in Discrete Optimization. In *OR*, pages 155–160, 2006.

[101] Karthik Shekhar, Petter Brodin, Mark M. Davis, and Arup K. Chakraborty. Automatic classification of cellular expression by nonlinear stochastic embedding (accense). *Proceedings of the National Academy of Sciences*, 111(1):202–207, 2014.

[102] Shai S. Shen-Orr, Robert Tibshirani, Purvesh Khatri, Dale L. Bodian, Frank Staedtler, Nicholas M. Perry, Trevor Hastie, Minnie M. Sarwal, Mark M. Davis, and Atul J. Butte. Cell typeâĂŞspecific gene expression differences in complex tissues. *Nature Methods*, 7(4):287–289, March 2010.

[103] Jared T. Simpson and Richard Durbin. Efficient de novo assembly of large genomes using compressed data structures. *Genome Research*, 22(3):gr.126953.111–556, December 2011.

[104] Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J. Jones, and Inanç Birol. ABySS: a parallel assembler for short read sequence data. *Genome research*, 19(6):1117–1123, June 2009.

[105] Pawel Smialowski, Dmitrij Frishman, and Stefan Kramer. Pitfalls of supervised feature selection. *Bioinformatics*, 26(3):440–443, February 2010.

[106] Reiji Teramoto. Balanced gradient boosting from imbalanced data for clinical outcome prediction. *Statistical applications in genetics and molecular biology*, 8(1), 2009.

[107] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J. Lennon, Kenneth J. Livak, Tarjei S. Mikkelsen, and John L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudo-temporal ordering of single cells. *Nature Biotechnology*, 2014.

[108] Rene L Warren, Benjamin P Vandervalk, Steven JM Jones, and Inanc Birol. Links: Scaffolding genome assemblies with kilobase-long nanopore reads. *bioRxiv*, 2015.

[109] L. Weng, D. Ziliak, H. K. Im, E. R. Gamazon, S. Philips, A. T. Nguyen, Z. Desta, T. C. Skaar, the Consortium on Breast Cancer Pharmacogenomics COBRA, D. A. Flockhart, and R. S. Huang. Genome-wide discovery of genetic variants affecting tamoxifen sensitivity and their clinical and functional validation. *Annals of Oncology*, March 2013.

[110] Hui Y. Xiong, Babak Alipanahi, Leo J. Lee, Hannes Bretschneider, Daniele Merico, Ryan K. C. Yuen, Yimin Hua, Serge Gueroussov, Hamed S. Najafabadi, Timothy R. Hughes, Quaid Morris, Yoseph Barash, Adrian R. Krainer, Nebojsa Jojic, Stephen W. Scherer, Benjamin J. Blencowe, and Brendan J. Frey. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218), 2015.

[111] Daniel R. Zerbino, Gayle K. McEwen, Elliott H. Margulies, and Ewan Birney. Pebble and rock band: heuristic resolution of repeats and scaffolding in the velvet short-read de novo assembler. *PloS one*, 4(12):e8407+, 2009.

[112] Yi Zhong, Yin-Wooi Wan, Kaifang Pang, Lionel M. L. Chow, and Zhandong Liu. Digital sorting of complex tissues for cell type-specific gene expression profiles. *BMC Bioinformatics*, 14:89, 2013.