Constrained De Novo Peptide Identification via Multi-objective Optimization

Malard J.M., Heredia-Langner A., Baxter D.J., Jarman K.H. and Cannon W.R.
Pacific Northwest National Laboratory, Richland WA, USA
Email: jm.malard@pnl.gov

Abstract

Automatic de novo peptide identification from collision-induced dissociation tandem mass spectrometry data is made difficult by large plateaus in the fitness landscapes of scoring functions and the fuzzy nature of the constraints that is due to noise in the data. Two different scoring functions are combined into a parallel multi-objective optimization framework.

1. Peptide identification

High-throughput proteomic techniques seek to characterize the state of the proteome in a cell population. A typical procedure may involve extracting cellular proteins followed by tryptic digestion and then separating the peptides with liquid chromatography. The separated peptides are then identified by tandem mass spectrometry (MS/MS). Ideally, peptides will subsequently be quantitated, post-translational modifications will be determined and the information regarding the peptides will be assembled into a picture of the proteomic state of a cell population. Accurate identification of peptides is critical for drawing biologically meaningful conclusions.

For this reason, there has been much work recently on developing peptide identification methods for MS/MS spectra. This area of research has proceeded on two fronts, the first of which seeks to take advantage of the wide availability of genome sequences. The database search methods try to identify the peptide that resulted in the observed MS/MS spectrum by picking the best candidate from a list of peptides generated from the genome sequence (e.g. Eng *et. al.* [8] and Perkins *et. al.* [22]). *De novo* methods on the other hand, seek to identify a peptide simply from the observed MS/MS spectrum (e.g. Dančík *et. al.* [6], Fernandez-de-Cossio *et. al.* [10], Jarman and Cannon [3,13] and Heredia-Langner *et. al.* [12]).

Generally, the work presented here focuses on an alternative to traditional graph theory-based *de novo* methods that additionally allows one to "jump-start" de-novo peptide identification using either existing peptide databases, or the results from database search methods such as SEQUEST [8]. The latter may be particularly useful when extended to the identification of post-translational modifications to peptides that are known to occur, but are currently hard to identify.

The specific question addressed in this report is simply put: how can one combine effectively different scoring functions and constraints in order to enable automatic de-novo peptide identification without resorting to a peptide database. The solution proposed here is based on population (or sampling) based multi-objective (or vector) optimization methods that combine the different scoring functions and constraints into non-commensurate objective functions. The result of one such optimization is a list of peptide sequences that best match the data under the assumption that the scoring functions cannot be ranked in order of importance or reliability. Surprisingly enough, this list is relatively small.

2.1. *De novo* peptide identification from tandem mass spectrometry data

Peptide identification via *de novo* methods are not widely recognized as an effective means to identify the best peptides from either first principles or from a short list of putative peptides. First, MS/MS spectra often do not contain enough information to allow for unambiguous determination of the entire peptide sequence. It has been estimated that 50% of spectra are missing enough peaks to allow only partial interpretation [15]. Second, *de novo* approaches can be computationally intensive, which is an important criterion for high-throughput proteomics. Third, graph algorithms that search the space of spectra peaks for putative peptide sequences can easily generate candidate sequences that fail to meet the constraints of the probabilistic model that underlies the scoring function. Yet, some of those trial sequences may on one hand contain information that is critical to the overall search, or on the other hand, they may have very high scores that hide the true peptide sequence. Still, there is a significant need for *de novo* sequencing methods because often the most biologically interesting peptides, such as those containing mutations and frame-shifts, may not be in the sequence database to begin with. This will be especially true in clinical or field settings where the genome of the organism being studied differs from the genome of the organism that was sequenced.

2.2. Two scoring functions

The problem of *de novo* peptide identification can be seen as one of numerical optimization, but one must keep in sight that the results of numerical optimization are only as meaningful as the underlying quantitative model or scoring function. It is assumed that this model is faithful to reality. Several models have been proposed for determining the likelihood of a match between the fragment ions in candidate sequence and an experimental mass spectrum, see for example [8,18,22,24,30,9,31,1,6,27,28,23]. In the context of biology, this creates a situation where many models exist that may lead to different predictions. Conflicts between predictions may be resolved through the development of more accurate models. A complementary approach is to develop optimization algorithms that are robust not only with respect to noise in the data but also to the quantitative models, leaving room for biologists to explore the implications of the various models and of their simplifying assumptions.

Two scoring functions are considered in this paper. Both are based on probabilistic models of peak matching, neither is particularly expensive to compute compared to the overall computation time. That is not to say that current computer architectures are well suited for them. In fact, both involve many more conditional branches and non-strided memory accesses than is found in typical number crunching production applications run on high-end computing platforms. Both scores take as input a sequence of amino acids and an electrostatic charge. A synthetic spectrum is generated that is then compared to the experimental spectra. The execution time of both functions is roughly linear in the number of peaks in both spectra.

The H-score function: the score defined by Heredia-Langner *et al* [12] is a weighted sum of terms that capture various aspects of peak matching. Only one of these terms is used here. That term quantifies how much two peptides have portions within them that are similar. The scoring algorithm sweeps the synthetic spectrum over the experimental spectrum and counts the matches, if any, are found. Computationally, the algorithm counts the number of different values in a table with as many rows and columns as are present in the synthetic and experimental spectra respectively.

The ϑ-score function: Jarman *et al* [13] define a likelihood ratio with respect to two explicit hypotheses H_0 and H_A. The null hypothesis H_0 is that spectral peaks match ion fragments purely by chance; the alternative hypothesis H_A is that spectral peaks match ion fragments because the candidate sequence is in the sample. The ϑ *fitness function* is the likelihood ratio between H_0 and H_A. Further details of this scoring function are described in [13]. The time needed to compute any one ϑ-score is small and although details are omitted here, they do not affect the conclusions of this work.

2.4 Optimization Constraints

When scoring function parameters are unconstrained, peptide identification can result in high-scoring sequences with little biological relevance. Constraints arise because of how tryptic digestion works and what this says about the matching peptide sequences one might expect to detect. First, the ratio m/z of mass to electrostatic charge of the parent peptide is known to some degree of accuracy along with the probable charge z. One also knows, that z is 1, 2 or 3. Second, it is also very likely that the peptide sequence will end in K or R. The constraints on charge and terminal are embodied in the sequence representation. This will be made precise in the next section.

Here, the mass constraint is embodied in the scoring function Δ of a weak model, parallel to H and ϑ. The Δ-score of a putative sequence s is simply the negative of the magnitude of the difference of the mass of s and that of the experimentally measured parent peptide, m. The mass of the parent peptide is determined by assuming a charge and calculating the mass from the observed mass-to-charge ratio (m/z). However since the parent mass is only know up to some accuracy, an additive threshold is used such that the Δ-scores of two sequences s and s' are considered equal whenever their difference doesn't exceed this given threshold. The idea of handling constraints in terms of objective functions is not new. Surry and Radcliffe [26] proposed one such approach, called Constrained Multi-objective Optimization by Genetic Algorithm (COMOGA), when a large fraction of generated trial solutions do not satisfy the many constraints of a complex industrial problem. COMOGA switches back and forth between single-objective and multi-objective optimization to try maintain a minimum number of feasible solutions. In the present work, the number of objective functions is constant throughout the evolution, however one of the objectives is used preferentially during the local searches that are described in the next section as Lamarckian evolution.

3. Multi-objective optimization

Recent advances in numerical computing and computer science have enabled the solution of large-scale single objective optimization problems with respect to millions of free variables in traditional quantitative sciences; see for instance [5,2]. Coello [4] and Deb [7] give a good overview of many current multiobjective optimization methods, including minimization of the weighted Lp norm of the vector of objective values, Goal Programming and Attainment methods and the ε–method. Evolutionary algorithms and in particular genetic algorithms do have the potential to utilize fully massively parallel computer clusters. Constraint handling and the availability of cheap parallel computing platforms are two reasons for looking at multi-objective optimization. There is at least one other reason, but before it is presented, some concepts from evolutionary computing need to be review first.

3.1. Genetic algorithms and population based optimization

The basic concepts of Genetic Algorithms are reviewed next, more details and specific techniques can be found in many good books and online courses including [19]. A genetic algorithm evolves a *population* of so-called *genotypes* through genetic operators such as selection, mutation and recombination. Those genotypes are called *sequences* in order to avoid confusion with the genomic data of Computational Biology. Beware that a *sequence* (in italics) need not be the same thing as an amino acid sequence; it may in fact represent a set of such sequences. Another source of confusion is that the values assigned to components of a *sequence* (a variable length vector) are called *alleles*; in the present context, these alleles would be the amino acids in putative peptide sequences. Whether or not *sequences* should be represented only has bit *sequences* or as some more congenial representation, such as character strings or integer vectors, is open to debate. Here we take the approach taken by many practitioners, that individual members of the population are vectors of unspecified atomic values. The basic constituents of a genetic algorithm are its representation, population (and structure), fitness, selection and reproduction (mutation and cross-over).

Representation: Each of the *sequences* represents nine pairs of parent charges and amino acid sequences. For instance, the *sequence RPNQTHL* represents the sequences RPNQTHL, RPNQTHLK, and RPNQTHLR with charges of 1, 2 and 3. In the language of Evolutionary Computing, those pairs of amino acid sequences and charges are called the *"phenotypes"* of the sequence. Again, in order to avoid confusion in this paper with existing biological terms, *"phenotypes"* will be called *realizations* of the corresponding *sequence*. Realizations came in Evolutionary Computing from the observation that the effectiveness of genetic algorithms to identify regions of the search space where a global optima may be found does not translate in general in an ability to actually pin-point that global optima. Thus, the *Darwinian evolution* enacted by the genetic algorithm is complemented by a *Lamarckian evolution* that may be implemented by a standard Newton-Raphson algorithm if first order derivatives are available. Those genetic algorithms are also known as *memetic algorithms*, see for example [20]. In this paper, a *sequence* is represented by a vector of integers between 0 and 18. The null *allele* is a placeholder; *alleles* between 1 and 18 encode amino acids amino acids having the same mass. There are 20 amino acids but the two pairs I/L and K/Q cannot be told apart from their masses alone. The scores of a *sequence s* are those of the sequence with the highest ϑ-score among the realizations of s. Any other score could serve for that matter; an unbiased approach more in line with COMOGA could be implemented instead, for instance by selecting a random scoring function for each *sequence* or for each generation.

Population: An initial population is generated, often entirely at random or by perturbation of some given sequences; here, the three best candidates found by SEQUEST. At each generation, a subset of the population is selected for reproduction based on their intrinsic or relative fitness. The new *sequences* then replace some other selection of *sequences* in the population. Successive generations are computed until their number exceeds some threshold, or the maximum fitness within the population stabilizes for long enough, or until say 95% of the population is within one standard deviation from the maximum, or until some other criterion is satisfied. The population is said to have *converged* when the distribution of fitness values among the population has stabilized in some way. It is important to note that the population size remains fixed throughout the generations.

The *structure* of a population imposes constraints on which *sequences* are allowed to mate together. Each *sequence* has a neighborhood, or *deme*, from within which mates can be selected. *Sequences* whose *demes* do not intersect cannot produce an offspring. In the *diffusion* models, individual *sequences* are assigned to nodes of a conceptual regular grid and the *deme* of a *sequence* consists of its nearest neighbors along that grid. In the island model, *demes* form a partition of the entire population, but random migrations of *sequences* at regular intervals ensure that the islands remain related. A population that consists of a unique island is called *panmictic*. The island and diffusion models were introduced in order to take advantage of parallel hardware. It was observed afterwards that structure could reduce the total number of sequences sampled by the genetic algorithm before it converges. Thus, population structure may speedup the convergence of serial algorithms as well. In this paper, each time a migration takes place a random cycle of all islands is computed. Each island sends one of its members to its left neighbor along the cycle and replaces that member by the *sequence* it receives from its right neighbor.

Fitness: The *fitness* and *value* are two related but separate concepts. The value of a sequence is its score with respect to the target spectra. The fitness of a sequence is how well it fits the data relative to all sequences represented in the population. Note the use of italics. The important concept here is that of *selection pressure* imparted by the fitness function. If one defines the distance between two peptide sequences to be the minimum number of changes one must impart to one in order to get the other one, the scoring functions that are considered in this paper have large variations over short distances as well as very small variations over long distances. Thus when selection is based entirely on scores a population consisting of low fitness *sequences* may stagnate. Alternatively, a single highly fit *sequence* may be irremediably lost after a single *"allele"* is modified. Typically, in single objective optimization, the raw scores are rescaled to yield a fitness function. In multi-objective optimization, the approach followed here is to rank the *sequences* based on the values of their realizations. The way this ranking is done is explained in the next section, the important point here is that the rank then becomes the value of the *sequence,* which is rescaled to produce a fitness value. In this paper, one score is arbitrarily chosen, say the ϑ-score and the H, ϑ and Δ-scores of a *sequence s* are the corresponding scores of that realization of *s* with the highest the ϑ-score. Other approaches are possible and not hard to implement.

Selection: The probability of a *sequence* being selected is typically not proportional to its fitness or score. Randomness is introduced in the selection process in order to preserve any *"genetic"* diversity within the population. With binary tournament selection, two *sequences* from the current population are chosen at random (with equal probability unrelated to fitness) but the *sequence* with the highest score is selected for reproduction. There are many variations on this theme, for instance more than two *sequences* might be chosen initially and more than one of those might be allowed to mate. A similar selection process can occur when new *sequences* replace existing *sequences*, although in practice a simpler "replace-worst" or "replace-random" rule is commonly enforced. Here 10% of the population is replaced at each generation.

Reproduction: Most genetic algorithms select pairs of parent *sequences*, say p_1 and p_2. *Unary* or 1-point *crossover* creates two new *sequences* c_1 and c_2 by interchanging randomly selected initial segments from p_1 and p_2. With *2-point crossover*, the new *sequences* are constructed by interchanging interior segments from the parent *sequences*. For example, if p_1 is *RPQTHLKPPN* and p_2 is *nfihtvvaha*, where case is not significant, a possible pair of children might be *RPQfihtvPN* and *nTHLKPvaha*. The children c_1 and c_2 are said to complement each other. Some genetic algorithms insert both children back in the population but this is not common. It is common however to perturb *sequences* before they are inserted back into the population. A typical *mutation* operator will randomly replace each *"allele"* from a *sequence* with a preset probability. In the present context, that probability is equal to the inverse of the maximum *sequence* length, and on average, only a few *"alleles"* are changed during a mutation.

3.2 Pareto dominance and ranking

A sequence s is said to dominate (in the Pareto sense) any other sequence s' such that the H, ϑ and Δ-scores of s are higher than that of s'. If those three scores represented the judgments of three panelists, the sequence s' might be discarded has having no merit. In any given population, those sequences that are dominated by no other sequence in that same population are called, not surprisingly, *non-dominated*. The real interest is of course for those sequences that are non-dominated within the entire space of sequences. Those are called dominant, or *efficient*, sequences.

It is common in practice to ascribe some hierarchy among fitness functions say by computing a weighted sum of the scores. One limitation here is that it can be very hard if at all possible to find a set of weights that is effective for a wide variety of peptide identification problems. Another limitation is that depending on the constraints and how they are handled, the weighted sum approach can miss some efficient *sequences*. This phenomenon is illustrated in Figure 1. The same genetic algorithm was run once over 1000 generations with an initial panmictic population of size 3000 using only the H-score to determine fitness. It was run a second time using the same experimental spectra but then using only the ϑ-score. Finally, the algorithm was run a third time by combining the H, ϑ and Δ-scores using Pareto ranking as described below.

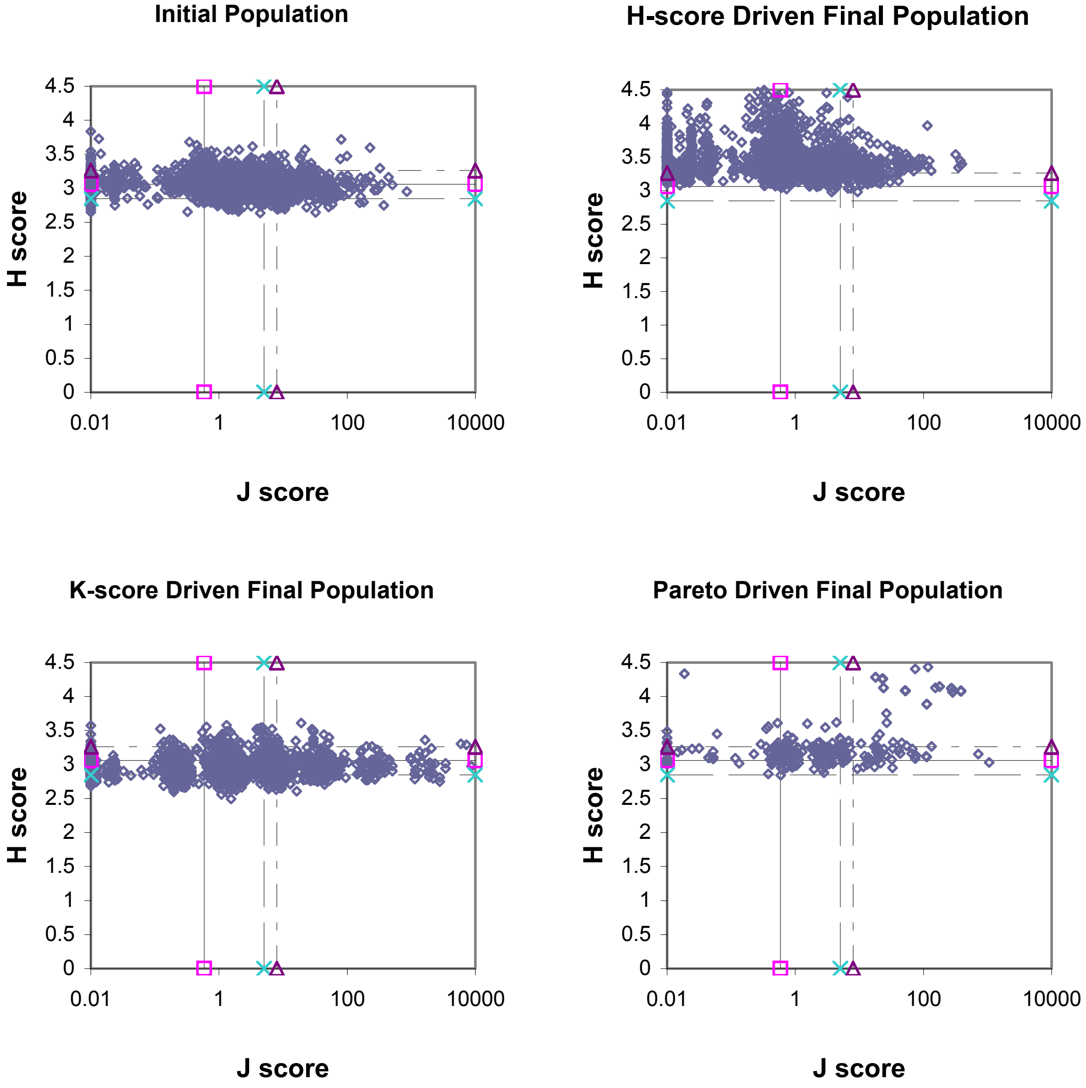

Figure 1 Distributions of H and ϑ-scores for different evolutionary paths of the same initial population.

Figure 1 displays the distribution of the H and ϑ-scores for the initial population and the final populations for each of the three runs. In all cases, the maximum *sequence* length is 40. The horizontal and vertical lines indicate the scores of the best (straight line and square symbol), second best (dashed line and star symbol) and third best (dot-dashed line and triangle symbol) scoring peptides according to SEQUEST. The H-driven and ϑ-driven populations see their fitness distributions migrating toward one side of their respective scatter plot. A linear combination of these two scores will migrate the score distributions also further away from the bottom left hand side corner but along a different axis.

Multi-objective algorithms based on Pareto dominance do not ascribe arbitrary importance to the different scores, they retain all that cannot be thrown out, but they also rely on the dominance relation to stir the overall population toward efficient *sequences* across generations. Ranking starts at zero for non-dominated *sequences* in the population. Different methods are employed to assigned higher ranks to the remaining *sequences*; see for instance [7]. The ranking used for this paper is due to Fonseca and Fleming, which is simply the number of *sequences* that dominate the given sequence. This rank can be computed within any one island by a single sweep of the *sequences* ascribed to that island. The rank is typically subtracted from the maximum rank to produce a fitness function that can be maximized. Table 1 illustrates this particular ranking algorithm using some sequences matched against the spectra from the sequence RNPQTHLKP.

Table 1 Fonseca-Fleming ranks for a small sample of peptide sequences matches against the spectra of a known peptide. The last three sequences are the SEQUEST candidates; the sequence in italics is actually the putatively correct one.

	Sequence	H	ϑ	Δ	Rank
1	RPNQTHLK	1.79	**76.13**	-114.5	0
2	RLPQTHNK	1.47	0.66	-114.5	2
3	RLPTQHPK	1.56	0.57	-131.5	0
4	*RNPQTHLKP*	**1.84**	23.33	**-17.4**	0

Looking at Table 1 it would be incorrect to conclude that one scoring function is better than the others, for one thing, in [13], the ϑ is only evaluated on sequences whose total mass is close enough to the target parent mass. Similarly, there are many peptides that have all the right amino acids, but in the wrong order and hence the right mass but very low H and ϑ-scores.

3.3 Parallel Computation

We used a modified version of PGAPack [16] from among several equally good toolkits for implementing parallel genetic algorithms. Paraphrasing Levine: "*PGAPack is a general-purpose, data-structure-neutral, parallel genetic algorithm toolkit for building parallel genetic algorithms based on the Message Passing Interface [21] library*". PGAPack is not a scripting language like RPL2 [25] and supports only panmictic population. The fact that PGAPack is public and its source is available made it possible to add support for both the island model and Pareto ranking. The parallel execution is very simple. MPI processes are ascribed to a single island throughout the evolution of the initial population. Each island owns a separate MPI communicator and operates as a task farm. A dedicated master process does all the bookkeeping and sorting required by the genetic algorithm and dispatches *sequences* to the other processes on the island. Pareto ranks are computed separately on each island by the corresponding master process. Global Pareto ranks across all islands could have been computed instead but would not have likely affected the scalability significantly at least for the Goldberg rank [17]. Migration occurs at regular intervals. A random cyclic path across all islands is computed and broadcasted by one of the master processes. Each master process then sends one *sequence* to its left neighbor along the cycle and this *sequence* is replaced by that received from the right.

2. Parallel Scalability and GA Efficiency

The interest of using parallel algorithms to search for peptide sequences that best match a single spectra comes from the fact that stochastic search algorithms such as genetic algorithms and Monte Carlo Simulations [11] can have both large warm up times and large overheads associated with the book-keeping of samples. Concurrent processes working together on the same experimental spectrum may reduce the warm up time they would require if they were assigned distinct spectra. This section addresses the question of the scalability of multi-objective genetic algorithms (GA) in the context of de novo peptide identification. All benchmarks ran on a Terascale HP cluster in the Molecular Science Computing Facility (MSCF) at Pacific Northwest National Laboratory. This cluster is composed of 1.5GHz Itanium 64-bit dual-processor workstations, linked together by a Quadrics QSNet 1 interconnect.

Table 2 illustrates how local populations can help reduce the bookkeeping time of the genetic algorithms while keeping the number of score evaluations constant. Table 2 shows the runtime in seconds for 100 generations of panmictic populations of various sizes evolved by a 2-process genetic algorithm. The time spent evaluating scores increases linearly with the number of score evaluations. It is small compared to the bookkeeping time of the algorithm that is seen to increase as the square of the population size.

Table 2 The cost in seconds of evaluating all 3 scores is seen to grow linearly with the population size while the overall runtime of evolving a panmictic population on a single SMP node across 100 generations rises much more sharply.

Population Size	Scoring	Total Time
512	0.51	22.21
1024	0.99	90.66
2048	1.99	349.02
4096	4.01	1384.62

Table 3 shows the time needed by a genetic algorithm to evolve an island structured population for 100 generations using the same peptide identification problem as in Table 2 and the same algorithmic parameters when the number of islands (one for each two processors) increases. The population on a single island decreases by a factor of two each time the number of island is doubled. Although the total number of score evaluated remains constant and although those populations remain connected to one another, the overall runtime is considerably reduced.

The last column of Table 3 shows the highest ϑ-score in the 100th generation of the respective population. That maximal score decreases as the number of islands increases and *"genetic"* diversity improves compared to a panmictic population. In particular, on a parallel computer where the solution time is not proportional to the total number of score evaluations, one will want in practice to maintain large enough island populations. Table 3 addresses the question of why not simply do peptide identifications of all the generated spectra, all at once in an embarrassingly parallel fashion. Sharing the identification tasks decreases the overheads and helps preserve diversity in the *sequence* population.

Table 3 For a constant total population size of 2048 *sequences* over 100 generations the reduction in runtime the cost of book-keeping and memory overheads is seen to decrease rapidly with the number of islands, which is half the number of processors. Times are in seconds.

p	Islands	Score time	Total time	*Seq.* per island	Best ϑ-score
2	1	1.99	349.29	2048	15639.77
4	2	0.98	89.87	1024	19781.35
8	2	0.66	89.52	1024	19781.35
16	2	0.29	88.49	1024	19781.35
8	4	0.50	23.99	512	6996.46
16	8	0.26	6.35	256	6996.46

4. Conclusions

A framework has been presented for combining de novo peptide identification methods. The distinctive feature of our approach, based on Pareto ranking, is that it can accommodate constraints and possibly conflicting scoring functions. We have also shown how population structure can significantly improve the wall clock time of peptide identification while at the same time maintaining some exchange of information across local populations. This paper does not address the question of the quality or biological relevance of the identified peptides, nor does it addresses the questions of the optimal number of islands, the optimal number of processors per islands etc; the latter are problem and hardware dependent.

Acknowledgment

This research was funded by the Office of Biological and Environmental Research in the U.S. Department of Energy under contract 41966A and performed in the William R. Wiley Environmental Molecular Sciences Laboratory at the Pacific Northwest National Laboratory. The EMSL is funded by the Office of Biological and Environmental Research in the U.S. Department of Energy. PNNL is operated by Battelle for the U.S. Department of Energy under Contract DE-AC06-76RLO 1830.

This research was performed in part using the Molecular Science Computing Facility (MSCF) in the William R. Wiley Environmental Molecular Sciences Laboratory, a national scientific user facility sponsored by the U.S. Department of Energy's Office of Biological and Environmental Research and located at the Pacific Northwest National Laboratory. Pacific Northwest is operated for the Department of Energy by Battelle.

5. Bibliography

[1] V. Bafna, N. Edwards, "SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database", *Bioinformatics*, 2001, 17, pp. S13-21.

[2] A. Bouaricha, J. Moré, “Impact of partial separability on large-scale optimization", *Comp. Optim. Appl.*, 1997, 7, pp. 27-40.

[3] W.R. Cannon, K.D. Jarman, "Improved peptide sequencing using isotope information inherent in tandem mass spectra”, *Rapid Commun. in Mass Spectro.*, 2003, 17, pp. 1793-801.

[4] C.A. Coello Coello, “An updated Survey of GA-based Multiobjective Optimization Techniques”, *ACM Computing Surveys,* 2000, 32(2), pp. 109-43.

[5] R. Conn, N.I.M. Gould, Ph.L. Toint, "Large-scale nonlinear constrained optimization: a current survey", in *Algorithms for continuous optimization: the state of the art*, 434, Kluwer Academic, (ed.) E. Spedicato, pp. 287-332, 1994.

[6] V. Dancík, T.A. Addona, K.R. Clauser, J.E. Vath, P. Pevzner, "De novo peptide sequencing via tandem mass spectrometry", *J. Comput. Biol.*, 1999, 6, pp. 327-42.

[7] K. Deb, *"Multi-Objective Optimization using Evolutionary Algorithms"*, John Wiley & Sons, Chichester, UK, 2001.

[8] K. Eng, A.L. McCormack, J.R.I. Yates, "An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database", Journal *of the American Society of Mass Spectrometry*, 1994, 5, pp. 976-89.

[9] J. Eriksson, B.T. Chait, D. Fenyo, "A Statistical Basis for Testing the Significance of Mass Spectrometric Protein Identification Results", *Analytical Chemistry*, 2000, 72, pp. 999-1005.

[10] J. Fernandez-de-Cossio, J. Gonzalez, Y. Satomi,T. Shima, N. Okumura, V. Beseda, L. Betancourt, G. Padron, Y. Shimonishi, T. Takao, "Automated interpretation of low-energy collision-induced dissociation spectra by SeqMS, a software aid for de novo sequencing by tandem mass spectrometry", *Electrophoresis*, 2000, 21, pp. 1964-9.

[11] G.S. Heffelfinger, M.E. Lewitt, "A comparison between two massively parallel algorithm for Monte Carlo computer simulation -- An investigation in the Grand Canonical Ensemble", *J. Comp. Chem.*, 1996, 17, pp. 250-65.

[12] A. Heredia-Langner, W.R. Cannon, K.D. Jarman, K.H. Jarman, "Sequence optimization as an alternative to de novo analysis of tandem mass spectrometry data", 2002, to appear in *Bioinformatics*.

[13] K.H. Jarman, W.R. Cannon, K.D. Jarman, A. Heredia-Langner, K.J. Auberry, and G.A. Anderson, "A statistical framework for peptide identification from tandem mass spectrometry data", submitted 2003.

[14] A. Keller, A.I. Nesvizhskii, E. Kolker, R. Aebersold, "Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search", *Analytical Chemistry*, 2002, *74*, pp. 5383-92.

[15] M. Kinter, N.E. Sherman, *Protein Sequencing and Identification Using Tandem Mass Spectrometry*, Wiley-Interscience, New York, 2000.

[16] D. Levine, "Users guide to the PGAPack parallel genetic algorithm library, Argonne National Laboratory, report ANL-95/18, January 1996

[17] J.M. Malard, "An application of singular values in protein binding", *International Workshop on Parallel Matrix and Applications*, Neufchatel CH, November 2002.

[18] M. Mann, M. Wilm, "Error-tolerant identification of peptides in sequence databases by peptide sequence tags", *Analytical Chemistry*, 1994, *66*, pp. 4390-9.

[19] Z. Michalewicz, *"Genetic Algorithms + Data Structures = Evolution Programs",* Springer-Verlag, 1994.

[20] P. Moscato, "Memetic algorithms: A short introduction", in *New Ideas in Optimization*, (eds.) Corne D., Dorigo M. and Glover F., McGraw-Hill, London UK, 1999, pp. 219-34.

[21] A. Geist, W. Gropp, S. Huss-Lederman, A. Lumsdaine, E. Lusk, W. Saphir, T. Skjellum, M. Snir, "*MPI-2: Extending the message-passing interface*", Argonne National Laboratory, Technical Report MCS-P568-0296, 1996.

[22] D.N. Perkins, D.J.C. Pappin, D.M. Creasy, J.S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data", *Electrophoresis*, 1999, *20*, pp. 3551-67.

[23] P.A. Pevzner, V. Dancik, C.L. Tang, "Mutation-tolerant protein identification by mass spectrometry", *Journal of Computational Biology*, 2000, 7(6), pp. 777-87.

[24] R. Sadygov, J.A. Yates, "A Hypergeometric Probability Model for Protein Identification and Validation Using Tandem Mass Spectral Data and Protein Sequence Databases", 2003, in press *Analytical Chemistry*.

[25] P.D. Surry, N.J. Radcliffe, "RPL2: A language and parallel framework for evolutionary computing", in *Parallel Problem Solving from Nature - PPSN III*, Eds. Y. Davidor, H-P. Schwefel, R. Männer, 1994, pp. 628-37.

[26] P.D. Surry, N.J. Radcliffe, "The COMOGA method: constrained optimisation by multiobjective genetic algorithms". *Control and Cybernetics*, 1997, 26(3), pp. 391-412.

[27] J.A. Taylor, R.S. Johnson, "Sequence database searches via de novo peptide sequencing by tandem mass spectrometry", *Rapid Commun. Mass Spectrom.*, 1997, 11, pp. 1067-75.

[28] J.A. Taylor, R.S. Johnson, "Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry", *Analytical Chemistry*, 2001, 73, pp. 2594-2604.

[29] D.A. Wolters, M.P. Washburn, J.R. Yates, "An automated multidimensional protein identification technology for shotgun proteomics", *Analytical Chemistry*, 2001, 73, pp. 5683-90.

[30] W. Zhang, B.T. Chait, "ProFound: An expert system for protein identification using mass spectrometric peptide mapping information", *Analytical Chemistry*, 2000, 72, pp. 2482-9.

[31] N. Zhang, R. Aebersold, B. Schwikowski, "ProbID: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data", *Proteomics*, 2002, 2, pp. 1406-12.