SIGEVOlution

newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation

Autumn 2008
Volume 3 Issue 3

in this issue

Editorial

GECCO has been recently ranked as the 11th most impactful conference in Artificial Intelligence / Machine Learning / Robotics / Human Computer Interaction. The ranking considered 701 conferences and took into account several factors such as the citation of papers, the quality of the referees' reports, the availability of resources to students, etc.

No doubt this is an amazing result for a conference that in Montréal will celebrate its 10th birthday. GECCO is just a kid, not even a teen, but it is positioned high on top near conferences that have been around for more than 20 years such as AAAI (the one with the highest impact among the 701 conferences included in the ranking), NIPS (ranked 2nd), and IJCAI (ranked 3rd). The people who envisioned GECCO and the people who made it what it is today scored an incredible result with their ten years of hard work and their long term commitment.

Some time ago I announced that the board was working on a little surprise. And here it is: an interview with John Holland. But this is just an appetizer, other interviews will follow. In Autumn, I worked with the board and other people from our community to select a list of 20 questions for a series of interviews with influential people, icons, of our fields. The interview with John Holland is the first of the series. In the next issues, the newsletter will host the interviews with the people who accepted the invitation.

After the interview, Clare Bates Congdon, H. Rex Gaskins, Gerardo M. Nava, and Carolyn Mattingly take us inside non-coding DNA to search for functional elements. Then, Julian Togelius reports on the simulated car racing competition which was held at the IEEE Symposium on Computational Intelligence and Games (CIG-2008) in Perth. If you like competitions, you should check the section about the five GECCO-2009 competitions since all the deadlines are still open!

At the end, I would like to thank the people who made this issue possible, John Holland, Lashon Booker, Clare Bates Congdon, H. Rex Gaskins, Gerardo M. Nava, Carolyn Mattingly, Julian Togelius, Wolfgang Banzhaf, Kalyan Deb, Dirk Thierens, Erik Goodman, Jeff Horn, Fernando Lobo, Eric Cantú-Paz, Riccardo Poli, Una-May O'Reilly, Rick Riolo, Franz Rothlauf, Marc Schoenauer, Darrell Whitley, Martin V. Butz, Xavier Llorá, Kumara Sastry, and board members Dave Davis and Martin Pelikan. Without them and without you reading, I would not be here writing. We are behind schedule by two issues, but we are working hard to catch up!

I hope you like the cover. I created it from a photo of John Holland I took during a workshop in Ann Arbor.

Pier Luca
June 14th, 2009

SIGEVOlution
Autumn 2008, Volume 3, Issue 3

Newsletter of the ACM Special Interest Group on Genetic and Evolutionary Computation.

Contributors to this Issue

John H. Holland
Lashon B. Booker
Clare Bates Congdon
H. Rex Gaskins
Gerardo M. Nava
Carolyn Mattingly
Julian Togelius

Contents

ISSN: 1931-8499

An Interview with John H. Holland

with an introduction by Lashon B. Booker

John H. Holland, Center for the Study of Complex Systems, University of Michigan, jholland@umich.edu

John Holland began his academic career as a graduate student at the University of Michigan studying with Arthur Burks, a designer of early programmable computers and colleague of John von Neumann. Under Burks' supervision, in 1959 Holland received what may have been the first Ph.D. in the world in the emerging field of computer science. He was subsequently hired as one of the first professors in the new department of Computer and Communication Science at the University of Michigan.

As the inventor of genetic algorithms, John established himself as one of the early pioneers in understanding the foundations of adaptation, learning, and modeling in both natural and artificial systems. His extensions of the genetic algorithm, first to a cognitive architecture called classifier systems and later to an ecological architecture known as Echo, have increased the reach of his ideas well beyond computer science. John Holland's imaginative, interdisciplinary approach to thinking about adaptation, emergence and complexity has made him one of the world's acknowledged leaders in the field of complex adaptive systems. In addition to his many intellectual achievements, John Holland has also been a valued mentor and friend for those of us who have had the privilege to be his students.

Holland is currently a Professor of Psychology and Professor of Electrical Engineering & Computer Science at the University of Michigan. He is also a member of the Board of Trustees and an External Professor at the Santa Fe Institute. Holland was made a MacArthur Fellow in 1992 and is a Fellow of the World Economic Forum.

Lashon B. Booker, Mitre Corporation

Q

Everybody knows the enormous influence you had in our field. Would you summarize the key ideas of genetic algorithms in 2-3 paragraphs for someone unfamiliar with the field?

Genetic algorithms (GAs) generate solutions by discovering and recombining building blocks (schemata) in a manner similar to the cross-breeding of natural organisms. Genetic Algorithms are at their best in finding improvements.

Q

What experiences in school, if any, influenced you to pursue a career in science?

From a very early age my parents played games (checkers and card games) with me. Once I began attending school I was almost automatically interested in "rule-based" systems, so science was immediately attractive. The major advances in physics while I was in high school certainly influenced me to study physics as an undergrad at MIT, and my encounter with Fisher's classic The Genetical Theory Of Natural Selection started me down the mathematical avenue that led to GAs.

Who are the three people whose work inspired you most in your research?

John von Neumann, Norbert Weiner, and Arthur Burks (through life-long interactions and mentorship).

What are the three books or papers that inspired you most?

The Organization Of Behavior by D. O. Hebb, The Genetical Theory Of Natural Selection by R.A. Fisher, and Theory Of Self-Reproducing Automata by J. von Neumann. The 1947 Moore School notes Theory And Techniques For Design Of Electronic Digital Computers greatly influenced my senior year at MIT.

As a founding father of this field, what is your own view about what genetic algorithms are? What did you expect them to be?

About this, see also my answer to the first question. And from the preface of Adaptation In Natural And Artificial Systems: “The possibility of ’intrinsic parallelism’ — the testing of many schemata by testing a single structure — is a direct offshoot of this approach.”

What are your favorite real-world applications of genetic algorithms?

Dave Goldberg’s early use of GAs to simulate control of gas pipeline transmission is still one of my favorites.

What is the biggest open question in the evolutionary computation area?

An important open question about GA-directed computations is the construction of models that exhibit the open-ended evolution we expect of natural systems such as ecosystems.

The combination of GAs and agent-based models offers the potential for an overarching theory of complex adaptive systems, particularly principled ways for discovering "lever points" in such systems.

Your books are sources of inspiration, is there any topic in your books which you hoped people would take more seriously?

There is still much to be learned about schemata and parallelism, particularly in the context of learning classifier systems.

Which ones are the most misunderstood/misquoted?

There is considerable misunderstanding of the role of schemata, despite many proofs of the theorem, including one by Feldman and Christiansen that uses the standard mathematical apparatus of mathematical genetics.

What new ideas are you working on and excited about?

The study of complex adaptive systems (cas) is challenging and endlessly fascinating for me. The treatment of language acquisition and evolution within the cas framework occupies much of my time, and a cas treatment of the co-evolution of “hierarchies” and signals is the central topic of the book I’m currently writing.

Q What books, tangentially related to the field, that you've read in the last year did you like the best?

Though I first read these books some time ago, they've had continuing influence: Das Glasperlenspiel by Herman Hesse, an unusually insightful view of what it means to construct a powerful overarching framework. The Name Of The Rose by Umberto Eco, one of the best description of deduction ever written. Labyrinths, by Jorge Luis Borges, for modeling and combinatorics. Tree And Leaf, by J.R.R. Tolkien, for language.

Q You had many successful PhD students, what is your recipe for PhD success?

First, find a broad question that REALLY interests you. Second, learn a lot about a lot of fields. Finally, find a "patron", a senior researcher who appreciates and understands your work, and will stand up for it, no matter how "far out" – i.e. original – it is.

Q Has thinking about evolution changed your view on things in general?

As we learn more about evolution, both computationally and biologically, I become more and more impressed with the scope of Darwin's original conception. One has only to read his Fertilisation Of Orchids to see the depth of his conception.

It's Not Junk!

The Search for Functional Elements in Noncoding DNA

Clare Bates Congdon, University of Southern Maine, Portland (ME), congdon@usm.maine.edu
H. Rex Gaskins, University of Illinois at Urbana-Champaign, Urbana (IL), hgaskins@uiuc.edu
Gerardo M. Nava, University of Illinois at Urbana-Champaign, Urbana (IL), gnavamo2@uiuc.edu
Carolyn Mattingly, Mount Desert Island Biological Laboratory, Salisbury Cove (ME), cmattin@mdibl.org

The human genome is approximately 3 billion basepairs long. An estimated 2-3% of DNA codes for genes; the remaining 97-98% is noncoding DNA [11]. Although the noncoding regions in DNA were once called "junk DNA" (with the assumption that these regions were not serving a purpose) it is now understood that within noncoding DNA are functional regions that affect the expression of genes [34]. However, we are still far from understanding the breadth of function in the noncoding regions, and identification of functional elements is a complex problem, difficult to study *in vitro* because of the enormous number of possibilities. In this project, we are searching *in silico* for candidate functional elements in noncoding DNA. These candidates will then be studied at the bench to assess function. Our guiding principle is that regions of noncoding DNA that have been conserved across evolutionary time are good candidates as functional elements; we use a genetic algorithms approach to search for these candidate elements, called motifs. Our system is thus called GAMI, Genetic Algorithms for Motif Inference. GAMI has been demonstrated to be a successful approach to this task, as will be described below.

The Big Picture

The human genome project and related sequencing efforts have resulted in vast amounts of DNA sequence data for humans and several other species. The development of computational techniques and tools for studying this sequence data is vital to increasing our understanding of the mechanisms of life.

A genome corresponds to the double stranded helix DNA (Deoxyribonucleic acid), and is represented as a linear sequence of four molecules called nucleotides or bases. The four bases found in DNA are adenine (A), cytosine (C), guanine (G) and thymine (T). The bases are naturally complementary, with an A in one strand of DNA pairing with a T in the other and C and G pairing with each other. Due to the pairing of bases, knowing the sequence of bases in one strand of DNA for an organism is sufficient to know the full genome and thus, a genome is reported as a single strand. However, the two strands function independently during the processes of transcription into RNA and then translation of the RNA into the proteins of the genes, and a particular gene occurs in a particular strand. Thus, functional elements in noncoding DNA may appear in either strand as well.

> *"it is now understood that within noncoding DNA are functional regions that affect the expression of genes"*

An example strand of noncoding DNA and its complement strand is illustrated in Figure 1. The DNA strand has an orientation, due to the bonds between the molecules. Each strand has a direction: The top strand is read from left to right and elements further to the left are said to be upstream while elements further to the right are said to be downstream. The bottom strand is read from right to left and the upstream is on the left and downstream is on the right.

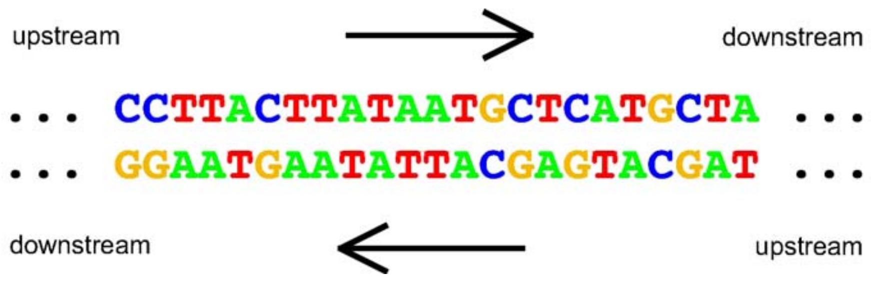

upstream
downstream
... CCTTACTTATAATGCTCATGCTA ...
... GGAATGAATATTACGAGTACGAT ...
downstream
upstream

Fig. 1: An example DNA sequence and its paired strand.

Within the part of a DNA strand that codes for a gene, there are alternating regions of exons and introns. The exons are the expressed DNA, which will be transcribed into RNA, and the introns are intervening sequence that are not ultimately translated into proteins. Thus, the introns are another form of noncoding DNA.

When DNA is in a single-stranded state, as it is during the processes of transcription into RNA (followed by translation of the RNA into the proteins of the genes), a chemical element called a transcription factor may bind to the noncoding regions, and influence the transcription process, enhancing or suppressing the expression of genes. Such a site is called a transcription factor binding site (TFBS). A striking example of the importance of noncoding regions is the relatively recent discovery that a 16-base-long element (named HACNS1) found in human DNA activates genes in the wrist and thumb and may be largely responsible for human's increased dexterity over chimps and monkeys [4].

Toxicological Significance

An underlying motivation in this work is to increase our understanding of the role that the environment plays in determining gene expression. Knowledge about the elements that regulate transcription will improve our ability to understand how environmental factors such as toxic metals (e.g. arsenic) interact with genes to influence gene expression. It is increasingly clear that environmental exposure plays a very important role in many biological processes that influence disease susceptibility. This is true for all common human diseases including cancer, cardiovascular diseases, and psychiatric disease, for example.

Genes of Interest

One of the genes that we are interested in studying is CFTR, the cystic fibrosis transmembrane regulator. This gene is mutated in individuals with cystic fibrosis and disrupts cross-membrane chloride transport [7]. In other words, cystic fibrosis interferes with the ability of cells to transfer salts in the membranes that line organs such as the lungs. (This causes the mucus in the lungs to become thick, leading to clogged airways and lung infections.) Interestingly, this gene is very well conserved through evolution, meaning that the sequence of bases in the coding regions of the gene has not varied much in some 500 million years [7]. In fish such as the kilifish, this gene helps species move between waters of varying salinity [26].

> *"[an element] found in human DNA activates genes in the wrist and thumb and may be largely responsible for human's increased dexterity over chimps and monkeys"*

In the work illustrated here, we use data reported in [27] for the cystic fibrosis transmembrane conductance regulator (CFTR). CFTR genomic sequence is available for approximately 40 organisms through the efforts of the National Human Genome Research Institute [27]. Nineteen species were used for the results illustrated here, based on the quality of the available upstream sequence from CFTR to the next known gene. The length of these sequences range from approximately 100 kb for species such as human down to 1813 bases in the fugu (a pufferfish) sequence. (The fugu genome is very short, relative to the human, so can be very helpful in honing in on candidate motifs.) The results reported later in this paper use the 2k upstream region (1813 bases for fugu).

The work reported here also investigates the glutamate-cysteine ligase catalytic subunit (GCLC), an environmentally responsive gene [33]. Deficiency in this gene in humans is associated with hemolytic anemia [2]. This gene is just one of a set of genes in a complex genetic pathway we are studying that may mediate the responsiveness of human cells to arsenic. The data set we are using here was obtained from the online Ensembl database [14] and curated by our team. In this data, the most primitive species is *Ciona intestinalis* (ciona), an invertibrate filter feeder; there are twelve target species in this data spanning from human to ciona. Upstream, downstream and intronic regions have been curated; the work reported here uses the 4kb upstream region.

DNA Motif Inference

Although the importance of the noncoding regions in now acknowledged, our ability to identify and predict the function of these DNA regions has been limited [3]. There are several studies that suggest that comparative analysis of evolutionarily diverse organisms will help predict functionally important noncoding regions [6], [2], [5].

The Motif Inference Task

As mentioned previously, one successful approach to computational motif inference in the noncoding regions is to look for patterns that appear to have been conserved through evolution; these may have been conserved because they are functional. This is the approach that we have taken. The candidate conserved elements are called motifs.

Input to the system. The data that we work with is comprised of comparable regions from the genomes of evolutionarily divergent species. For example, in the CFTR study reported here, we work with sequences that are immediately upstream from the CFTR gene in each species. Although DNA is typically represented in a linear sequence, it is tightly packed within three-dimensional space, and elements do not need to be proximal in the linear sense to the gene to be functional. In our current stage of research, we often look for functional elements in the upstream region of the genes, and sometimes use the upstream region up to the next known gene. In CFTR with human genome, this is approximately 100k bases long. When we do not have that long stretch of quality sequence to work with, or when we are doing a more focused study, we may look a specific distance upstream for all species, for example, 4k upstream from the gene. (We also work with downstream and intronic regions.)

The species that we work with depend in part on the species that have been sequenced for the genes being studied. CFTR is a particularly well studied gene, so there is more high quality sequence data available. Less well studied genes do not always have high quality DNA sequence data available for the noncoding regions of a variety of species. Thus, our data sets currently range in size from 40 species with sequences up to 100k base pairs (bp) long down to five species with sequences 1k bp. (The algorithm does not limit the length or number of sequences; runtime is roughly linear in respect to each of these.)

Output from the system. The result of the motif inference process is one or more motifs that appear to be strongly conserved across the input sequences. For the purposes of this work, a motif is defined in a given data set of nucleotide sequences as a pattern that occurs at least once in each sequence. A base pattern of length N is called an N-mer. Imperfect matches are expected; that is, the pattern might not be represented exactly in one or more of the input sequences. N-mers that are more strongly matched across the set of sequences are considered stronger motifs.

Specific Goals of Our Work

The specific goals of motif inference vary for different researchers. The following characteristics describe our unique combination of requirements for GAMI:

1. We are looking specifically for conserved regions, so it is important that there is support for the motifs identified in each of the sequences in the dataset.
2. We are looking specificially in noncoding regions, where there is generally less conservation than in coding regions.
3. We are looking for candidate functional regions and not specifically for TFBS.
4. We want to be able to search in long sequence lengths, perhaps 100 kb or longer.
5. We want to be able to search a large number of sequences when quality sequence is available.
6. We want to be able to search for large motifs, perhaps 100 bases or longer.

Several of these characteristics differ from other motif inference approaches. Many motif inference approaches do not require that motifs are contained in all of the sequences; this is evidenced in published benchmarking data sets such as [30]. Some tools search only for TFBSs and are therefore limited in the scope of regulatory elements that can be identified [5]. Some motif inference projects look in the core promoter region only (e.g., 1-200 bp upstream). Many motif inference projects restrict the number of sequences to a small number of sequences or short sequences due, in part, to runtime concerns. GAMI is not restricted by most of these limitations for reasons described below.

Approaches to Motif Inference

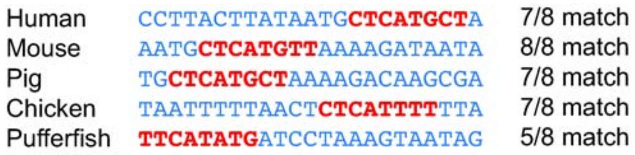
Lones and Tyrell [18] provide an excellent review of motif discovery, focusing on evolutionary computation as a specific approach. In particular, the work of Corne, Meade, and Sibley [12, 22] uses a system design similar to GAMI. As noted by Lones and Tyrell, the most common approach to locating and characterizing conserved regions in sets of biological sequences is to first use a global-sequence-alignment system, such as CLUSTAL [28] or PipMaker [25] for genomic sequence. Global sequence alignment is computationally expensive, particularly as the number and length of the sequences increases. Furthermore, for evolutionarily distant species, the degree of sequence divergence precludes global alignment, especially in noncoding regions.

> *“we sought a more computationally tractable approach, and search the space of possible motifs ”*

Another highly favored approach to motif inference is to search for an optimized and co-adapted set of window locations across the set of sequences. The set of windows form a matrix that describes the motif. This approach is used, for example, in MEME [1], Gibbs Sampler [29], Fogel et al. [15], and GEMFA [3]. The latter two also use evolutionary computation approaches. With GAMI [9, 8], we sought a more computationally tractable approach, and search the space of possible motifs instead of the space of possible matrices. Additional recent evolutionary computation approaches in the literature include [20], which uses data clustering to distributed the evolving population across the search space, and [19], which extends the method to co-evolve Boolean rules that describe the relationships among the evoling motifs; [6], which explores the combination of a window-location approach and the consensus motif approach; and [17], which uses a multi-objective GA, using motif length and strength as the competing objectives.

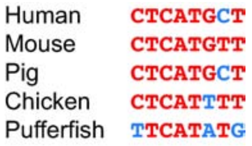
GAMI System Design

GAMI [9, 8] is an approach to motif inference based on genetic algorithms. GAMI searches a set of DNA sequences for patterns that appear at least once in each sequence. The motif representation is the standard consensus motif: an N-mer composed of the bases A, C, G, and T. For example, if we are searching for 8-mers, possible motifs identified would include CATGCAAT, TAGGAACT, ACTTACGT, etc.



Human	CCTTACTTATAATG**CTCATGCT**A	7/8 match
Mouse	AATG**CTCATGTT**AAAAGATAATA	8/8 match
Pig	TG**CTCATGCT**AAAAGACAAGCGA	7/8 match
Chicken	TAATTTTTAACT**CTCATTTT**TTA	7/8 match
Pufferfish	**TTCATATG**ATCCTAAAGTAATAG	5/8 match

Fig. 2: An example of the 8-mer motif CTCATGTT matching example data; the motif location is shown in red. The overall MC score for this motif in this data is 7+8+7+7+5=34 out of a possible 40. (A total of six possible base matches have been missed.)



Human	CTCATGCT
Mouse	CTCATGTT
Pig	CTCATGCT
Chicken	CTCATTTT
Pufferfish	TTCATATG

Fig. 3: The motif locations from Figure 2 aligned. The bases that do not match the motif are shown in blue.

As the initial fitness function, we used a metric we call “match count” (MC). To evaluate the MC of a given motif, each sequence is searched to find the best consecutive match for that motif within that sequence. Forward and reverse-complement matches are considered for each sequence. The best match maximizes the number of bases that match the motif across all the sequences; there might be more than one best match for a given motif and nucleotide sequence (but this does not alter the score). A match for the motif CTCATGTT in example data is shown in Figure 2. The (maximum) number of bases matched in each sequence is the score for that motif with that sequence; the score for the motif across all sequences in the data is the overall score for the motif. When illustrated in this form, this is evocative of the window-based search algorithms, which are searching for a set of window locations that work best together. However note that GAMI is looking for the strongest pattern, which can then be mapped onto the best locations, whereas the window-based approaches are looking for the best locations, which can then be expressed as a pattern. Figure 3 illustrates the best match for the motif locations in each sequence in an aligned form.

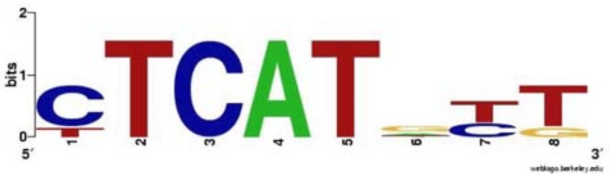


Fig. 4: The motif locations from Figure 2 as a logogram.

Figure 4 (created with Weblogo [13]) illustrates the motif as a logogram [24], a standard graphical depiction of an aligned set of motifs. The height of each column is a measure of the information content in that column, with a maximum value of 2 bits. This motif illustrates the strong "core region" of TCAT (with more variance in other positions), which is typical of TFBS and other functional elements.

The Genetic Algorithms Design

Once the problem of motif inference is defined as a search through a set of possible strings and a fitness function is defined, the application of a GA to the problem is straightforward. We have implemented a standard GA following the structure of Genesis [16], and using the ACGT alphabet, two-point crossover, point mutation, and generational replacement with roulette-wheel selection.

GAMI searches the space of N-mers in the alphabet A, C, G, T; the length of the N-mers is currently fixed at runtime. Initially, the MC metric described above was used at the fitness function.

Modifications to the standard GA include:

1. A new mutation operator was added that truncates one end of a motif and adds a random base to the opposite end. This is called "slide mutation", as it has the effect of sliding a motif one base to the left or to the right.
2. A new local-search mutation operator was added that chooses a position of the motif at random, checks the score for all four bases in that position, and sets the position to the base yeilding the highest score (ties are broken at random). This is called "directed mutation".
3. Elitism is used to save the highest scoring solutions from one generation into the next.
4. A seeding operation for the initial population has been added that seeds a percentage of the initial population by sampling from a specific sequence in the input.

We typically run with a population size of 1000 and an elitism rate of 50%, and thus, the result of a run is 500 high-scoring candidates for functional elements. Seeding has proven very successful in jump-staring the search process, greatly reducing the needed search time.

A Discussion of the Search Space and Representation

In comparing GAMI to the window-based approach, it is important to look at the size of the search space as well as the representational power of the system.

The Size of the Search Space. The search space for GAMI is the number of possible N-mers, so the size of the search space is based on the length of the N-mers: 4^N. For example, with 10-mers, the size of the search space is 4^{10} or approximately 1 million; with 20-mers, the size of the search space is 4^{20} or on the order of 10^{12}.

The search space for the window-location approach is the number of possible combinations of window locations within each sequence. Thus, the size of the search space is based primarily on the length of the sequences in the data. If all the sequences are of length L, there are roughly L places to position the window within each sequence; if there are S sequences in the data set, the number of possible solutions is on the order of L^S. For example, with a relatively small dataset of 5 sequences, each 300 bp long, the number of possible solutions is 300^5, approximately 2.43×10^{12}.

Because the sizes of the search spaces depend on different factors (length of the N-mers vs. length and number of sequences in the data set), the best way to compare the size of the search space for the two representations is to look at specific examples. For example, the *GCLC* data set contains 12 sequences that are each 4000 bp long; the GAMI search space is still 4^{20}, or roughly 10^{12} while the size of the search space for a window-based approach is 4000^{12}, or roughly 10^{43}.

Evaluation Time. Although the size of the search space is markedly smaller with GAMI than with a window-based approach, the evaluation time per candidate solution is markedly higher.

In GAMI, the evaluation time varies with the length of the sequences in the data, the length of the motif, and the number of sequences, as the motif must be compared against every position along the sequence. In a window-based approach, the evaluation time varies only with the length of the motif and the number of sequences in the data. For example, with the *GCLC* upstream regions, GAMI requires on the order of 12×20×4000 comparisons to evaluate a potential solution, while a window-based approach requires on the order of 12×20 comparisons.

One might also look at the complexity of the overall search process. Again with the example *GCLC* data, the size of the search space is roughly 10^{43} for a window-based approach, times a factor of 12×20 to evaluate all those solutions; with GAMI, the size of the search space is 10^{12}, times a factor of $12 \times 20 \times 4000$ to evaluate all those solutions.

Representational Power. Clearly, the GAMI representation is unable to represent solutions with the same degree of resolution as a window-based approach. However, a GAMI match against the data can still be represented as a logogram, as is common in many motif-inference programs. There may be tasks where an extended alphabet (such as the IUPAC nucleotide code) is appropriate, but the 4-base alphabet is not a limitation of the system (the alphabet can easily be extended). Of course, extending the alphabet also increases the size of the search space, so these tradeoffs need to be weighed carefully.

Successes and Results

GAMI is designed to be able to infer relatively long motifs, e.g., 20-50 bases long. In our initial work, we focused on looking for 20-mers as a good starting point. For example, in early work, [9], we used GAMI to search for 20-mers on several datasets, including non-coding regions of the CFTR gene. as well as the CYP3A7, CYP2U1 and CYP2U2 genes, which are also responsive to environmental cues. In this work, we were able to demonstrate GAMI's abilities to Identify highly conserved 20mers, looking at upstream, downstream, and intronic regions in CFTR and finding candidate conserved elements ranging from 82 to 91 percent conserved. This work included replicating the results of Fogel [15] for finding known TFBSs in data. The Oct and NF-kb data sets from Fogel were used to confirm GAMI's ability to find known TFBS; GAMI found the TFBS in both data sets. These data sets were also used to compare GAMI's results to exhaustive search, and verified that GAMI found all the highest scoring solutions found by exhaustive search.

In more recent work [8], we have also assessed the ability of our scoring metric (match count) to capture highly conserved regions, particularly as compared with information content, the metric more typically used for motif inference. For these studies, we curated data from the upstream region of the SOX21 gene for divergent species ranging from human (*Homo sapiens*) to fugu (*Takifugu rubripes*). These sequences were 3 kb long and contained two ultra-conserved regions confirmed experimentally [31] for functional importance in zebrafish embryos. This work demonstrated that information content does not capture highly conserved regions as effectively as match count does.

This work further investigated the CFTR data as well as upstream regions from the environmentally responsive cytosolic glutathione transferase (GST) genes, and illustrated that motifs identified by GAMI correlate with known TFBSs in the TRANSFAC database [21]. While this does not confirm functionality in these contexts, the identification by GAMI of known TFBSs does indicate further promise for GAMI as an approach for identifying functional regions.

Some Observations and 20-20 Hindsight

Recall that GAMI is not dependent on sequence alignments and is not deterred by the lengths of the sequences or the number of species, which can both lead to computational bottlenecks in alignment-based and matrix-based approaches to motif inference. We ran our early experiments with five sequences, following the conventions observed in other motif inference programs, later to realize that running with more species is preferable for our specific goals.

We have recognized that the probability of finding a motif that appears to be conserved but is due only to random chance increases with the length of the sequences in the dataset. For example, there are 4^{10} different possible 10mers on the 4-character ACGT alphabet, approximately 1 million. Thus, in a single short 10bp sequence, the probability of matching a given motif is approximately 1 in 1 million, while in a random sequence of length 100k, the probability of finding a specific 10mer motif is 1 in 10. While the probability of finding a specific 10mer motif in a data set of 5 sequences, each of length 100k is $(1/10)^5$ or one in 100,000, the probability of finding any one of the possible 10mer motifs in that data is ten to one (we would expect to find 10 10mers in common to the 5 100k sequences).

The probability of finding a 10mer in common to five randomized 10k sequences drops to 1/10,000, but these numbers are for motifs that are 100 percent conserved. The probability of finding apparent conservation in randomized sequences rises dramatically as the degree of conservation drops. Again, shorter sequences, more sequences, and longer motifs decrease the probability of inferring a putative motif that is due to coincidence, rather than conservation, while longer sequences, fewer sequences, and shorter motifs increase the probability.

Due to these observations, we now realize that we should always run experiments with as many high-quality sequences as we have available, and that there may be advantages to doing some studies with shorter sequences lengths than we have available. In particular, shorter regions immediately upstream of genes have been better studied, so are more likely to have previously verified functional elements. While our goal is to find novel functional elements, the confirmation of known functional elements has been useful for verifying the utility of our algorithm.

Similarly, there are more previously verified elements that are shorter (e.g., 10bp long). Thus, we have found it useful to conduct some of our verification work with 10-mers although there is little reason to use a genetic algorithms search with this formulation when looking for 10mers because there are only 1,048,576 possible 10-mer motifs on the ACGT alphabet and it is possible to generate and test them all.

Recent Results with CFTR Data

The population-based strategy of the GA and the use of elitism along with preventing duplicate solutions from being saved means that we have multiple high-scoring solutions as the result of a run. When we looked closer at these solutions, we found that the best solutions tended to “overlap”. For example, if the best 10mer was TTTTAACCTG, found in the human sequence at position 1925, the second best 10mer might be TTTAACCTGC, found in the human sequence at position 1926. The overlapping of motifs is helpful for inferring longer motifs than those explicitly searched for. However one may also wish to look for something like “the 10 best motifs that do not overlap”. Some of our recent experiments have thus explored the set of best motifs that can be found while allowing the motifs to overlap in only one base.

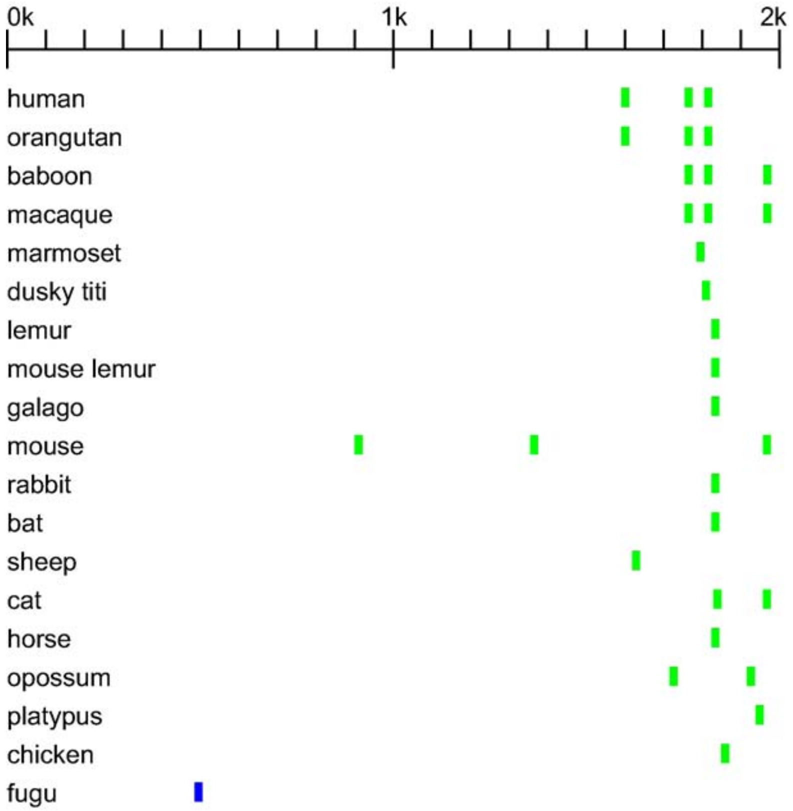


Fig. 5: The motif GGGAAGGAGG within each 2 kb sequence in the CFTR dataset.

Figure 5 illustrates some results for searching the 2k upstream region of the CFTR gene. In this run, we looked for 10mers in the 19 species shown, and looked in particular at the 10 best motifs found, allowing only one basepair overlap in the human sequence. This result is presented in [10]. The top of the figure illustrates the 2k human sequence upstream from the CFTR gene. The green rectangles indicate the approximate location of the motif match in each species; the fugu rectangle is blue, indicating that the best match appeared in the opposite strand of DNA. (Note that the width of the rectangles is not to scale for greater visibility.) This motif is GGGAAGGAGG, and is the eighth best motif in this experiment, 94.7% conserved across the 19 species. The logogram for this motif is also shown in Figure 6 for reference.

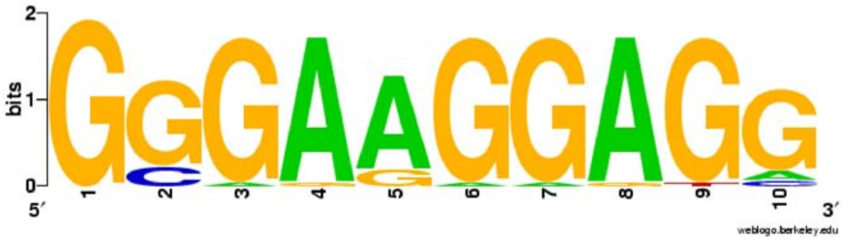


Fig. 6: The motif GGGAAGGAGG in the 2kb CFTR dataset as a logogram.

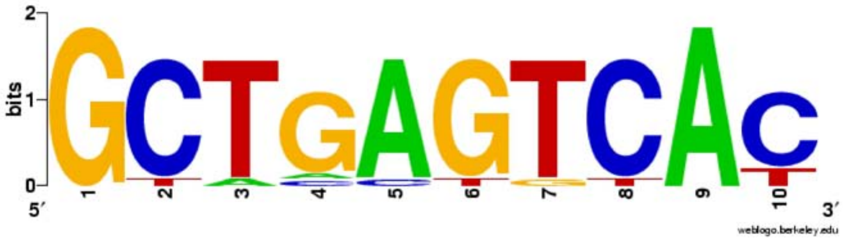


Fig. 7: Example of the TFBS found by GAMI in the upstream region of the GCLC gene that binds to the Nrf2 and small Maf transcription factors in humans.

Of particular note in this output is the high degree of positional conservation. While there is nothing in the GAMI algorithm to prefer motifs with positional conservation, the fact that it appears in this motif does increase our interest in this as a candidate functional element. In addition to the location that tends to be in the area of 200 bp from the gene (which would be on the right), the triplet pattern in the four highest-order species is also of great interest.

In the past year, there has been increasing annotation and validation of transcription factor binding sites (TFBS) in the TRANSFAC database [21] of the genes we have been studying, allowing us to further assess GAMI's ability to identify actual functional elements.

Recent Results with GCLC Data

In the GCLC data, GAMI identified the motif GCTGAGTCAC as a putative functional element that is 93.3% conserved across the 12 species. New comparisons of GAMI results to TRANSFAC found that this motif corresponds exactly to the core region of the ARE4 TFBS in humans (this corresponds to TRANSFAC identifier HS$GCLC_08, accession number R22708), which binds to the Nrf2 and small Maf transcription factors. Figure 7 illustrates the motif found. These transcription fators are responsible for for regulating pathways involving GCLC[32] . Thus, this is an exciting confirmation of GAMI's abilities.

Looking Toward the Future

Since most elements controlling eukaryotic genes function in regulatory modules rather than in isolation, there has been increasing interest in developing computational systems to infer these modules [23]. In addition to capturing a more complete description of the mechanisms underlying gene expression, computationally derived candidate modules may have higher predictive value than candidate motifs alone [23]. As we move toward adding this capability to GAMI, the studies with the 2k CFTR upstream sequences has again been informative.

Working with the 10 best motifs as illustrated in Figure 5, we have assembled candidate motifs by hand to illustrate the module concept. The result is illustrated in Figure 8. This figure shows the locations of three distinct motifs (colored red, blue, and green) in the 10 higher species of the CFTR data set. Again, for each of these motifs individually, the GAMI fitness function has no preference for positional conservation when scoring motifs; this is an emergent property.

This module is composed of the two highest scoring motifs, TGGGTGGGGG (green) and TGCCCAGGTT (blue), both 96.8% conserved across the 19 species, and the fourth highest scoring motif, GGAAGGAGCG (red), 95.8% conserved. The presence of candidate modules among the highest scoring motifs in the CFTR data is an intriguing development.

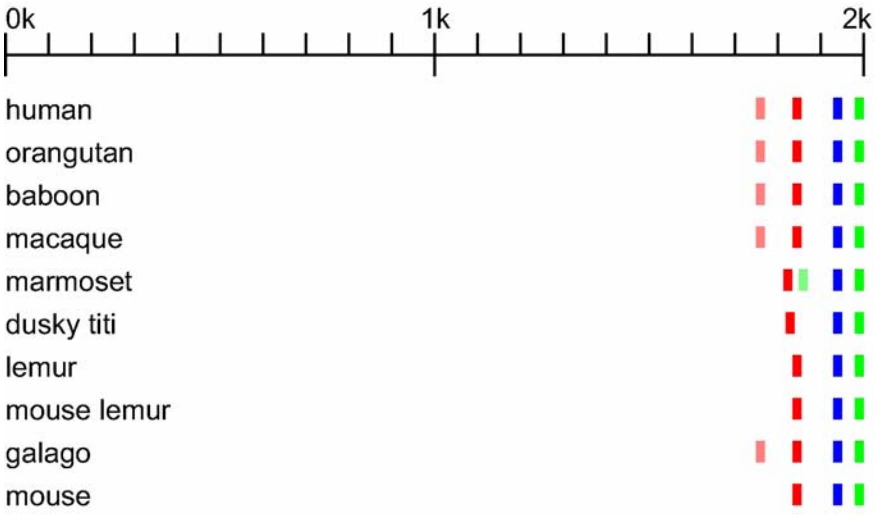


Fig. 8: The location of three of the highest scoring motifs on the higher species in the 2k upstream CFTR dataset. The stronger colors illustrate a putative module; the pastel colors are additional matches of the motif to the sequence.

Discussion and Conclusions

GAMI is a successful approach to the inference of conserved functional elements in noncoding DNA, designed to be able to handle larger tasks in terms of longer sequences, more sequences, and longer motifs in linear time. As a genetic algorithms search, another strength of GAMI is that it is able to find multiple candidate motifs in a single run.

GAMI has been validated for several genes with known functional elements and has discovered many candidate elements that are not yet characterized. Again, due to the GA search, the approach is malleable, and it is easy to modify the system, for example, changing the fitness function to prevent overlaps.

It is also easy to extend the alphabet used for the motifs, allowing a range of wildcard characters as well and an extension to proteins that we have been exploring.

We are working to investigate the effects of running GAMI on artificial datasets with implanted motifs at different levels of degeneracy to gain an understanding of the effects of degeneracy on GAMI's abilities to infer motifs. As part of these experiments, we will compare GAMI to competing approaches such as MEME [1] and Gibbs Sampler [29].

A limitation of the system includes the current fixed-width motif length. We recognize that it may be preferable to provide GAMI with the ability to express variable-length motifs and allow GAMI to determine the ideal length of motifs for a dataset as part of its selection process. This, however, is not clearly problematic in that the algorithm currently successfully identifies the regions of interest within the genome if not the specific lengths of these regions of interest.

GAMI does not currently consider the position of the motif in each sequence in the data set in the fitness function. It may be desirable to add this as an option for users. However, this is not clearly desirable. Currently, we are more likely to ask this question in a post processing stage rather than during evolution. This is because positional similarity is not required for similar function, in part due to the three-dimensional structure of the DNA within the cell. Elements that may appear to be distal in the standard two-dimensional representation of the DNA sequence may in fact be in similar positions relative to the coding regions in three-dimensional space.

We have also begun work to use GAMI for the inference of functional elements for co-expressed genes. In this task, the input data consists of similar regions across species as before, but also data for multiple genes is in the data set (each sequence is a sequence relating to a particular gene as well as a particular species). On the surface, this is a direct application of the algorithm; however, in this case it is desirable to let genes drop out of consideration for a match because the fact that multiple genes are active or inactive in the same environments does not mean that they are responding to the same biochemical signals. Instead, it may be that the product of one gene is influencing another in the co-expressed set of genes.

As previously discussed, the next major development in this line of research will be to address the inference of modules, rather than isolated elements.

Acknowledgments

The authors would like to thank Joseph Aman, Brian Leighton, Charles Fizer, Noah Smith, Rachel Teo, and Junes Thete for their contributions to this project; Eric Green and Gary Fogel for sharing data; and D. Lunn A. Sawyer, John Kuehne, Steve Bryant, and Roy McMorran for technical support.

This project was supported by P20 RR-016463 NIH/NCRR, as well as a Pilot Project (New Investigator Award) from the MDIBL Center for Membrane Toxicity Studies ES03828 NIEHS/NIH.

Bibliography

[1] Timothy L Bailey and Charles Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36. AAAI Press, Menlo Park, CA, 1994.

[2] E Beutler, R Moroose, L Kramer, T Gelbart, and L Forman. Gamma-glutamylcysteine synthetase deficiency and hemolytic anemia. *Blood*, 75(1):271–3, Jan 1 1990.

[3] Chengpeng Bi. A genetic-based EM motif-finding algorithm for biological sequence analysis. In *Proc. 2007 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)*, pages 275–282, 2007.

[4] Ewen Callaway. Junk dna may have handed us a gripping future. *New Scientist*, September 2008.

[5] K. Cartharius, Kornelie Frech, Korbinian Grote, B. Klocke, M. Haltmeier, Andreas Klingenhoff, Matthias Frisch, M. Bayerlein, and Thomas Werner. Matinspector and beyond: promoter analysis based on transcription factor binding sites. *Bioinformatics*, 21(13):2933–2942, 2005.

[6] T.-M. Chan, K.-S. Leung, and K.-H. Lee. TFBS identification by position- and consensus-led genetic algorithm with local filtering. In *Proceedings of the 2007 Genetic and Evolutionary Computation Conference (GECCO 2007)*, pages 377–384, July 2007.

[7] TY Chen and TC Hwang. Clc-0 and cftr: chloride channels evolved from transporters. *Physiol Rev.*, 88:351–87, 2008.

[8] C. B. Congdon, J. Aman, G. M. Nava, H. R. Gaskins, and C. Mattingly. An evaluation of information content as a metric for the inference of putative conserved noncoding regions in DNA sequences using a genetic algorithms approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, January, 2008.

[9] C. B. Congdon, C. W. Fizer, N. W. Smith, H. R. Gaskins, J Aman, G M Nava, and C Mattingly. Preliminary results for gami: A genetic algorithms approach to motif inference. In *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB-2005)*, pages 97–104. IEEE Press, 2005.

[10] C. B. Congdon, H. R. Gaskins, G. M. Nava, and C. Mattingly. Towards interactive visualization for exploring conserved motifs in noncoding dna sequence. *Proceedings of the 2007 IEEE Frontiers in the Convergence of Bioscience and Information Technologies (FBIT 2007)*, October, 2007.

[11] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431:931–45, 2004.

[12] D Corne, A Meade, and R Sibly. Evolving core promoter signal motifs. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 1162–1169. IEEE Press, 2001.

[13] G E Crooks, G Hon, J M Chandonia, and S E Brenner. WebLogo: A sequence logo generator. *Genome Res*, 14:1188–1190, 2004.

[14] V Curwen, E Eyras, T D Andrews, L Clarke, E Mongin, S M Searle, and M Clamp. The ensembl automatic gene annotation system. *Genome Res*, 14:942–50, 2004.

[15] G B Fogel, D G Weekes, G Varga, H B Harlow, J E Onyia, and C Su. Discovery of sequence motifs related to co-expression of genes using evolutionary computation. *Nucleic Acids Res*, 32(13):3826–3835, 2004.

[16] J. J. Grefenstette. A user's guide to GENESIS. Technical report, Navy Center for Applied Research in AI, Washington, DC, 1987. Source code updated 1990.

[17] Mehmet Kaya. Motif discovery using multi-objective genetic algorithm in biosequences. In *Proc. 2007 Int. Symposium on Intelligent Data Analysis (IDA 2007)*, volume 4723 of *LNCS*, 2007.

[18] Michael A Lones and Andy M Tyrrell. The evolutionary computation approach to motif discovery in biological sequences. In *Genetic and Evolutionary Computation Conference (GECCO-2005); Workshop on Biological Applications of Genetic and Evolutionary Computation*. ACM SIGEVO, 2005.

[19] Michael A. Lones and Andy M. Tyrrell. A co-evolutionary framework for regulatory motif discovery. In *Proceedings of the 2007 Congress on Evolutionary Computation (CEC2007)*. IEEE, 2007.

[20] Michael A. Lones and Andy M. Tyrrell. Regulatory motif discovery using a population clustering evolutionary algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(3):403–414, 2007.

[21] V Matys, E Fricke, R Geffers, E Gossling, M Haubrock, R Hehl, K Hornischer, D Karas, A E Kel, O V Kel-Morgoulis, D U Kloos, S Land, B Lewicki-Potapov, H Michael, R Munch, I Reuter, S Rotert, H Saxel, M Scheer, S Thiele, and E Wingender. Transfac: Transcriptional regulation, from patterns to profiles. *Nucleic Acids Res*, 31:374–378, 2003.

[22] A Meade, D Corne, and R Sibly. Discovering patterns in microsatellite flanks with evolutionary computation by evolving discriminatory DNA motifs. In *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 1–6. IEEE Press, 2002.

[23] N Pierstorff, CM Bergman, and T Wiehe. Identifying cis-regulatory modules by combining comparative and compositional analysis of dna. *Bioinformatics*, 22(23):2858–64, 2006.

[24] Thomas D. Schneider and R. Michael Stephens. Sequence logos: A new way to display consensus sequences. *Nucl. Acids Res.*, 18, 1990.

[25] S Schwartz, Z Zhang, K A Frazer, A Smit, C Riemer, J Bouck, R Gibbs, R Hardison, and W Miller. Pipmaker–a web server for aligning two genomic dna sequences. *Genome Res*, 10(4):577–06, Apr 2000.

[26] GR Scott, DW Baker, PM Schulte, and CM Wood. Physiological and molecular mechanisms of osmoregulatory plasticity in killifish after seawater transfer. *J Exp Biol.*, 211(Pt 15):2450–9, 2008.

[27] J W Thomas, J W Touchman, R W Blakesley, G G Bouffard, S M Beckstrom-Sternberg, and E H Margulies. Comparative analyses of multi-species sequences from targeted genomic regions. *Nature*, 424:788–793, 2003.

[28] J D Thompson, D G Higgins, and T J Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22), 1994.

[29] W Thompson, E C Rouchka, and C E Lawrence. Gibbs recursive sampler: finding transcription factor binding sites. *Nucleic Acids Res*, 31(13):3580–3585, 2003.

[30] M. Tompa, N. Li, T.L. Bailey, G.M. Church, B. De Moor, E. Eskin, A.V. Favorov, M.C. Frith, Y. Fu, J.W. Kent, V.J. Makeev, A.A. Mironov, W.S. Noble, G. Pavesi, G. Pesole, and M. Ry. An assessment of computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1):137–144, January 2005.

[31] A Woolfe, M Goodson, D K Goode, et al. Highly conserved non-coding sequences are associated with vertebrate development. *PLoS Biol*, 3(e7):116–130, 2005.

[32] H Yang, N Magilnick, C Lee, D Kalmaz, X Ou, JY Chan, and SC Lu. Nrf1 and nrf2 regulate rat glutamate-cysteine ligase catalytic subunit transcription indirectly via nf-kappab and ap-1. *Mol Cell Biol*, 25:5933–5946, 2005.

[33] P Yang, WR Bamlet, JO Ebbert, WR Taylor, and M de Andrade. Glutathione pathway genes and lung cancer risk in young and old populations. *Carcinogenesis*, 25(10):1935–44, 2004.

[34] E Zuckerkandl and G Cavalli. Combinatorial epigenetics, "junk dna", and the evolution of complex organisms. *Gene.*, 390:232–42, 2007.

About the authors

Clare Bates Congdon earned her PhD in Computer Science and Engineering from The University of Michigan, and is now an Assistant Professor in the Computer Science Department and Chief Scientific Officer of Bioinformatics and Intelligent Systems in the Research Computing Group at the University of Southern Maine. Her research interests include evolutionary computation approaches as applied to bioinformatics, specifically sequence analysis and phylogenetics; she also works in evolutionary agents and evolutionary art.

H. Rex Gaskins obtained the Ph.D. degree in nutritional sciences with a research focus in cell biology from The University of Georgia in 1989. From 1989-92, he completed postdoctoral studies in mouse genetics at The Jackson Laboratory in Bar Harbor, Maine. He is now a Professor at the University of Illinois at Urbana-Champaign with appointments in the Departments of Animal Sciences and Pathobiology, the Division of Nutritional Sciences, and the Institute for Genomic Biology. Research in his laboratory focuses on host-intestinal microbiota interactions relevant to inflammatory disorders, redox regulation of cell fate, and comparative genomics of redox homeostasis and inducible xenobiotic metabolism.

Carolyn J. Mattingly earned her Ph.D. in molecular toxicology from Tulane University. She is Director of the Bioinformatics Department at the Mount Desert Island Biological Laboratory in Salisbury Cove, Maine. Her research interests include development of the publicly available Comparative Toxicogenomics Database (CTD; http://ctd.mdibl.org) and discovery of regulatory motifs in non-coding regions of ATP-binding cassette (ABC) genes using cross-species comparative genomic analyses.

Gerardo M. Nava obtained DVM and MS degrees from the Universidad Nacional Autonoma de Mexico (UNAM) in 2000 and 2002, respectively. He is currently a PhD student at the University of Illinois at Urbana-Champaign and his research focuses on host-intestinal microbiota interactions relevant to inflammatory disorders and comparative genomics of redox homeostasis and inducible xenobiotic metabolism.

The Simulated Car Racing Competition @ CIG-2008

Julian Togelius, IDSIA, Switzerland
Conference webpage: http://www.csse.uwa.edu.au/cig08/
Competition webpage: http://cig.dei.polimi.it/

Overview of CIG-2008

The IEEE Symposium on Computational Intelligence and Games was held in Perth, Western Australia on December 15-18 2008. Over the past four years, this symposium has become the focal point for a community of researchers interested in applying computational intelligence techniques (especially evolutionary computation and neural networks) to games. "Games" in the conference title refers to traditional mathematical games studied in game theory, board games and last but absolutely not least video games.

Recently, efforts have been made to foster more participation in the conference and associated research community from the games industry. Both parties would arguably benefit from this, with researchers being informed about new interesting problems to work on (e.g. automatic environment generation and matching of players in online multiplayer games) and industry developers being informed about promising new techniques from the CI community. This effort was echoed in the selection of keynote speakers, with Jason Hutchens and Penny Sweetser giving industry perspectives on research opportunities for CI in games, complementing Jonathan Schaeffer's academic success story about solving checkers.

The conference, which was organized by general chairs Luigi Barone and Phil Hingston, managed to attract enough submissions that 53 high-quality papers could be selected for publication. In addition, three two-hour tutorials ("Learning to play games" by Simon Lucas, "Inducing Agent Models from Examples" by Bobby Bryant and "Measuring and Optimizing Player Satisfaction" by Georgios Yannakais and myself) and three competitions (The 2k Bot Prize, Simulated Car Racing and Ms. Pac-Man) were held as part of the conference.

Simulated Car Racing Competition

The simulated car racing competition was organized by Daniele Loiacono, Pier Luca Lanzi and myself, and was a follow-up on the similar competition associated with WCCI 2008. This competition in turn was a follow-up to the simulated car racing competions at CIG and CEC 2007, with the crucial difference that this competition was built around the TORCS open-source racing game rather than the the less sophisticated Java game that had been used previously.

The competition consisted in learning, or otherwise developing, a controller that raced a car as fast as possible around a previously unseen racing track. In the final event, the best three controllers competed on the same track simultaneously. This meant that in addition to finding and maintaining a good racing line, overtaking and collision avoidance were crucial skills to win the race.

For the competition, a "server-bot" was created for TORCS, allowing cars to be controlled through an external program via a UDP connection. Interfaces and example controllers were released in Java and C++, allowing competitors to easily connect their code to the competition environment. Additionally, the Java client came bundled with example learning algorithms. The setup of the interface is such that a number of "first-person" sensors representing what can be seen from the car is available from the controller, which has to supply commands for steering, accelerating, braking and gear shifting 50 times per second.

A total of five competitors entered the competition. Three of the competitors, Matt Simmerson, Luigi Cardamone and Aravind Gowrisankar, had used the NEAT neuroevolution algorithm by Kenneth Stanley to design their controllers. NEAT evolves neural network topologies and weights through a process known as complexification. The differences between these three competitors were therefore not so much in the learning algorithm, but in what sensors were fed to the neural network, how the outputs were interpreted and the training regime (what tracks were used and which, if any, opponents were present).

Diego Perez submitted a controller based on a ruleset evolved with a GA. Finally, Chung-Cheng Chiu submitted a controller based on a number of hard-coded heuristics, and which was not trained using any machine learning algorithm.

In the initial warm-up stage, each controller raced alone for 10000 game tics, approximatively 3 minutes and 20 seconds of actual game time. The three drivers that covered the more distance qualified for the next stage, the actual race: Luigi Cardamone's NEAT-based controller, Chung-Cheng Chiu's hard-coded controller, and the winner of the WCCI-2008 competition, developed by Matt Simmerson. The final event used the F1 scoring system and pitted the three controllers selected in the warm-up against each other, by running ten races on three different tracks. The controller submitted by Luigi Cardamone scored the most points, bestowing upon Luigi the title of Winner of the CIG 2008 Car Racing Competition. Chung-Cheng Chiu's controller came out as a close second, and the winner of the WCCI competition a rather distant third.

A number of observations can be made based on these results and from watching the actual races. To start with, progress seems to have been made in controller design between this competition and the last one, as two of the submitted controllers drove much better than the winner of the last competition. It is however surprising that a submission based on human ingenuity alone, without the help of any learning or optimization process, scored second place; this points out that we have much to learn about how to best apply evolutionary and other methods to controller design. For example, a large part of the performance difference between the controllers trained with NEAT is attributable to on which tracks they were trained; if they were not trained on any tracks similar to those used in the final scoring (as was the case with Matt Simmerson's controller) they tended to perform quite poorly.

When watching the movies of the final scoring event (available online at http://cig.dei.polimi.it) one is struck by that even though the controllers generally do a good job of driving fast and staying on track, they almost universally lack overtaking and collision avoidance skills. This is likely an effect of most controllers having been trained without other cars present on the track, or only with cars that drive too fast. Therefore, we hope to see entrants in the 2009 Car Racing Championship that have been trained in the presence of other controllers, perhaps using competitive coevolution. However, most of all we want to see as many, good and interesting controllers as possible submitted to this next iteration of the competition. Maybe something from you?

GECCO-2009 Competitions

The GECCO-2009 competition program is sponsored by NVIDIA and includes five competitions on solving the Rubik's cube, evolutionary art, GPU programming, learning and optimization.

Solve Rubik's Cube!

Parabon Computation, the leading on-demand computation utility, will sponsor a $2,000 prize competition in which contestants are challenged to use the company's Frontier Grid Service to evolve a program that can solve an arbitrarily scrambled Rubik's Cube in the minimal number of twists. Computer scientists have previously devised several strong solver algorithms; this competition aims to demonstrate that grid-scale computation and evolutionary processes can do just as well, or better!

Detailed information about the competition is available at:

http://parabon.com/news-events/gecco/rubiks-cube-contest.html

Important Dates

- Submission deadline: June 22nd 2009
- Conference: July 8th-12th 2009

Evolutionary Art

This competition aims at showing how genetic and evolutionary computation can be applied to create great artworks. The competition will award the best piece of evolved artwork (being a painting, a music score, a video, etc.) and the best system that exhibits some form of independent creativity.

Entrants must submit (1) a brief artist statement illustrating the concept, (2) a two page paper describing the technical details, and (3) a set of multimedia files to illustrate the result of the evolutionary process. Artists can either submit five still images, or a video up to 5 minutes, or a sound file, up to 5 minutes. All the submissions should be sent to lanzi@elet.polimi.it by June 26th.

The submissions will be evaluated by a panel of researchers from the evolutionary computation community and experts from art galleries who will score the submissions on several criteria including, the ability to demonstrate on-going novelty within some fixed criteria and medium (e.g., line drawing, image creation), technical and creative innovation, etc. The panel will select five finalists whose work will be shown during the conference. The attendees will vote to select the winner.

Important Dates

- Submission deadline: June 28th 2009
- Conference: July 8th-12th 2009

Organization

- Luc Courchesne, Université de Montréal
- Christian Gagne, Université Laval
- Pier Luca Lanzi, Politecnico di Milano
- Jon McCormack, Monash University

GPUs for Genetic and Evolutionary Computation

The GPU competition focuses on the applications of genetic and evolutionary computation that can maximally exploit the parallelism provided by low-cost consumer graphical cards. The competition will award the best applications both in terms of degree of parallelism obtained, in terms of overall speed-up, and in terms of programming style.

Entrants must submit (1) the application sources with the instructions to compile it and (2) a two page description of the application. Submissions will be reviewed by a committee of researchers from the evolutionary computation community and from industry. Each reviewer will score the submission according to 12 criteria concerning the submitted algorithm, the speed-up it achieves, and its impact on the evolutionary computation community. The total score will be obtained as the weighted sum of the 12 separate scores.

Submissions should be mailed to gecco2009@gpgpgpu.com no later than June 23, 2009. The final scores will be announced during GECCO.

Scoring

Submissions will be reviewed by a panel of researchers from the evolutionary computation community and from industry who will score each submission according to the following criteria.

Algorithm (50% of the total score)

Novelty	10%	Does the algorithm exploit the GPU in a novel way? (e.g., not just for fitness evaluation?)
Efficiency	10%	Does the algorithm efficiently use the GPU?
GPU-side	10%	How much of the algorithm is implemented GPU side?
Elegance	5%	Is the algorithm simple, easy to understand?
Portability	5%	Is the code parameterized for different GPU architectures and/or across vendors?
Suitability	10%	Does it use features of the GPU architecture logically and to the advantage of the program?

Speed (20% of the total score)

Speedup	10%	How much is the speed up compared to a well coded CPU version?
Resources	5%	What is the resource utilization? (Ideally a program should use the 100% of the GPU).
Scalability	5%	Will it scale? E.g. to new hardware, multiple GPUs, GPUs with fewer/more processors?

Evolutionary Computation (30% of the total score)

Utility	10%	Do the results benefit the EC/GA/GP community?
Practicality	10%	Were the results practically obtainable without GPU acceleration?
Science	10%	Is the system used to generate better quality science? For example, increasing statistical significance, increasing coverage of test cases or demonstrating greater generalization.

Important Dates

- Submission deadline: June 23rd 2009
- Conference: July 8th-12th 2009

Organizers

- Simon Harding, Memorial University of Newfoundland, Canada
- David Luebke, NVIDIA
- Pier Luca Lanzi, Politecnico di Milano
- Edmondo Orlotti, NVIDIA
- Antonino Tumeo, Politecnico di Milano

Simutated Car Racing
Contest 1: Learning to Drive

The first contest involves the design of a controller for a racing car that will compete on a set of unknown tracks first alone (against the clock) and then against other drivers. The controllers perceive the racing environment through a number of sensors that describe the relevant features of the car surroundings (e.g., the track limits, the position of near-by obstacles), of the car state (the fuel level, the engine RPMs, the current gear, etc.), and the current game state (lap time, number of lap, etc.). The controller can perform the typical driving actions (increasing the gear, accelerate, break, steering the wheel left or right, etc.)

The contest involves three Gran Prix on three (unknown) tracks. Each Gran Prix is organized in two stages: the warm-up and the actual race. In the warm-up, each driver will race alone for 10000 game tics, approximatively 3 minutes and 20 seconds of actual game time. The eight drivers that will cover the greatest distance will qualify for the next stage, the actual race. In the second stage, the eight drivers will race together. Each race consists of ten trials. The goal of each trial is to complete five laps from a randomly generated starting grid. At the end of each trial, the drivers will be scored using the F1 system: 10 points to the first controller that completed the three laps, 8 points to the second one, 6 to the third one, 5 to the fourth, 4 to the fifth one, 3 to the sixth, 2 to the seveth, and 1 to the eighth. The driver performing the fastest lap in the race will get two additional points. The driver completing the race with the smallest amount of damage will receive two extra points. The final score for each driver in the Grand Prix will be computed as the median of the 10 scores collected during the trials.

Important Dates

- Submission deadline: July 1st 2009
- Conference: July 8th-12th 2009

Competition Software

The competition software, including the servers for Linux & Windows, the C++ and Java clients, can be downloaded from the competition webpage:

http://cig.dei.polimi.it/?page_id=79

For inquiries send an email to championship2009@ieee-cig.org or visit the Car Racing Google Group at

http://groups.google.com/group/racingcompetition

Organizers

- Daniele Loiacono (Politecnico di Milano)
- Julian Togelius (IDSIA)
- Pier Luca Lanzi (Politecnico di Milano)

Simutated Car Racing
Contest 2: Optimizing Car Setup

The second contest simulates the days before a race when mechanics and pilots work on the car setup to find the one which will result in the best performance. The goal is to build an evolutionary algorithm that can replace the team of mechanics and pilots and can find the best car setup (e.g., gear ratio, wing area and angle, spring setup) on a given track.

The contest involves three tracks. The evolutionary algorithm will have to find the best car setup for each one of the tracks. The contest is divided into an optimization phase and an evaluation phase. During the optimization phase, the evolutionary algorithm will be applied to search for the best parameter setting. During the evaluation phase, the best solution will be scored according to the distance covered in a fixed amount of game time (or game tics).

A parameter setting is represented by a vector of real numbers. The competition software provides an API to evaluate a specific parameter setting on a track and returns the best lap time, the top speed, the distance raced, and the damage suffered. Through the API, it is possible to specify the amount of game tics to use for evaluating a car setup. The game tics spent for an evaluation are subtracted from the total amount of game tics available. When the 10 millions of game tics are exhausted or the evaluation process has taken up more than 2 hours of CPU time, no further evaluation will be possible.

Important Dates

- Submission deadline: July 1st 2009
- Conference: July 8th-12th 2009

Competition Software

The competition software, including the servers for Linux & Windows, the C++ and Java clients, can be downloaded from the competition webpage:

http://cig.dei.polimi.it/?page_id=79

For inquiries send an email to championship2009@ieee-cig.org or visit the Car Racing Google Group at

http://groups.google.com/group/racingcompetition

Video Tutorials

- Installation: http://www.vimeo.com/3852922
- Setup: http://www.vimeo.com/3852860

Organizers

- Luigi Cardamone (Politecnico di Milano)
- Daniele Loiacono (Politecnico di Milano)
- Julian Togelius (IDSIA)
- Pier Luca Lanzi (Politecnico di Milano)

New Issues of Journals

Evolutionary Computation 17(2) (www)

- Editorial Introduction Marc Schoenauer pp iii–iii (pdf)
- Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications Dimo Brockhoff, Eckart Zitzler pp 135–166 (pdf)
- A Genetic System Based on Simulated Crossover: Stability Analysis and Relationships with Neural Nets Marco Carpentieri pp 167–201 (pdf)
- Pair Approximations of Takeover Dynamics in Regular Population Structures Joshua L. Payne, Margaret J. Eppstein pp 203–229 (pdf)
- Adaptive Cellular Memetic Algorithms Nguyen Quang Huy, Ong Yew Soon, Lim Meng Hiot, Natalio Krasnogor pp 231–256 (pdf)
- A Strategy with Novel Evolutionary Features for the Iterated Prisoner's Dilemma Jiawei Li, Graham Kendall pp 257–274 (pdf)

Evolutionary Intelligence 1(4) (www)

- **Evolution of internal dynamics for neural network nodes**, David Montana, Eric VanWyk, Marshall Brinn, Joshua Montana and Stephen Milligan
pp 233-251 (pdf)
- **Genetic-based approach for cue phrase selection in dialogue act recognition**, Anwar Ali Yahya and Abd Rahman Ramli, pp 253-269 (pdf)
- **Automated feature selection in neuroevolution**, Maxine Tan, Michael Hartley, Michel Bister and Rudi Deklerck pp 271-292 (pdf)

Genetic Programming and Evolvable Machines 10(2) (www)

- **Incorporating characteristics of human creativity into an evolutionary art algorithm**, Steve DiPaola and Liane Gabora, pp 97-110 (pdf)
- **Using enhanced genetic programming techniques for evolving classifiers in the context of medical diagnosis**, Stephan M. Winkler, Michael Affenzeller and Stefan Wagner pp 111-140 (pdf)
- **Dynamic limits for bloat control in genetic programming and a review of past and current bloat theories**, Sara Silva and Ernesto Costa pp 141-179 (pdf)
- **A review of procedures to evolve quantum algorithms** Adrian Gepp and Phil Stocks pp 181-228 (pdf)
- Book Review: **Riccardo Poli, William B. Langdon, Nicholas F. McPhee: A Field Guide to Genetic Programming**, Lulu.com, 2008, 250 pp, ISBN 978-1-4092-0073-4 — Michael O'Neill pp 229-230 (pdf)

Natural Computing 8(2) (www)

- **Nature-inspired learning and adaptive systems**, Bogdan Gabrys and Davide Anguita pp 197-198 (pdf)
- **Integrative connectionist learning systems inspired by nature: current models, future trends and challenges**, Nikola Kasabov pp 199-218 (pdf)
- **A framework for machine learning based on dynamic physical fields**, Dymitr Ruta and Bogdan Gabrys pp 219-237 (pdf)
- **A survey on metaheuristics for stochastic combinatorial optimization**, Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella and Walter J. Gutjahr pp 239-287 (pdf)

- **Negative correlation in incremental learning**, Fernanda Li Minku, Hirotaka Inoue and Xin Yao pp 289-320 (pdf)
- **Solving the subset-sum problem with a light-based device**, Mihai Oltean and Oana Muntean pp 321-331 (pdf)
- **Successful preparation and analysis of a 5-site 2-variable DNA library**, Susannah Gal, Nancy Monteith and Anthony J. Macula pp 333-347 (pdf)
- **Competition and evolution in virtual plant communities: a new modeling approach**, Stefan Bornhofen and Claude Lattaud pp 349-385 (pdf)
- **Autopoiesis, the immune system, and adaptive information filtering**, Nikolaos Nanas and Anne de Roeck pp 387-427 (pdf)

July 2009

GECCO 2009 - Genetic and Evolutionary Computation Conference

July 8-12, 2009, Montréal, Canada

Homepage: http://www.sigevo.org/gecco-2009

Author notification: March 11, 2009

Camera-ready: April 22, 2009

The Genetic and Evolutionary Computation Conference (GECCO-2009) will present the latest high-quality results in the growing field of genetic and evolutionary computation.

Topics include: genetic algorithms, genetic programming, evolution strategies, evolutionary programming, real-world applications, learning classifier systems and other genetics-based machine learning, evolvable hardware, artificial life, adaptive behavior, ant colony optimization, swarm intelligence, biological applications, evolutionary robotics, coevolution, artificial immune systems, and more.

Organizers

General Chair:	Franz Rothlauf
Editor-in-Chief:	Günther Raidl
Business Committee:	Wolfgang Banzhaf Erik Goodman Una-May O'Reilly
Publicity Chair:	Martin Pelikan
Workshops Chair:	Anna I. Esparcia
Competitions Chairs:	Pier Luca Lanzi
Tutorials Chair:	Martin V. Butz
Late Breaking Papers Chair:	TBA
Local Chair:	Christian Gagné
EC in Practice Chairs:	David Davis Jörn Mehnen
Graduate Student Workshop Chair:	Steve Gustafson
Undergraduate Student Workshop Chair:	Frank Moore Clare Bates Congdon Larry Merkle

Venue

Delta Centre-Ville hotel is located in the heart of downtown, where Old Montreal and new Montreal blend seamlessly, and adjacent to vibrant nightlife, boutique shops and eclectic cuisine. For more information on Delta Centre-Ville, please visit:

www.deltahotels.com/hotels/hotels.php?hotelId=35

Visiting GECCO-2009 will be a great opportunity to visit the famous Montreal Jazz Festival (July 2-12, 2009):

www.montrealjazzfest.com/Fijm2008/festival_en.aspx

GECCO is sponsored by the Association for Computing Machinery Special Interest Group for Genetic and Evolutionary Computation.

September 2009

CIG-2009
IEEE Symposium on Computational Intelligence & Games
September 7-11, 2009 Politecnico di Milano, Milano,Italy - www.ieee-cig.org

IEEE Symposium on Computational Intelligence and Games (CIG-2009)

September 7-10, 2009, Milan, Italy

Homepage: http://www.ieee-cig.org

Submission deadline: June 14th, 2009

Competition papers: July 6th, 2009

Aim and Scope

Games are an ideal domain to study computational intelligence methods. They provide cheap, competitive, dynamic, reproducible environments suitable for testing new search algorithms, pattern based evaluation methods or learning concepts. At the same time they are interesting to observe, fun to play, and very attractive to students. This symposium, sponsored by the IEEE Computational Intelligence Society aims to bring together leading researchers and practitioners from both academia and industry to discuss recent advances and explore future directions in this field.

Topics of interest include, but are not limited to:

- Learning in games
- Evolutionary Computation for games
- Neural-based approaches for games
- Fuzzy-based approaches for games
- Console and video games
- Character Development and Narrative
- Opponent modeling in games
- CI/AI-based game design
- Multi-agent and multi-strategy learning
- Comparative studies
- Applications of game theory
- Board and card games
- Economic or mathematical games
- Imperfect information and non-deterministic games
- Evasion (predator/prey) games
- Realistic games for simulation or training purposes
- Player satisfaction in games
- Games for mobile or digital platforms
- Games involving control of physical objects
- Games involving physical simulation

Conference Committee

General Chair:	Pier Luca Lanzi
Program Chair:	Sung-Bae Cho
Proceedings Chair:	Luigi Barone & Philip Hingston
Publicity Chair:	Julian Togelius
Competition Chair:	Simon Lucas
Sponsorship Chair:	Georgios N. Yannakakis
Local Chairs:	Nicola Gatti and Daniele Loiacono

Important Dates (tentative schedule)

Paper submission:	14th June 2009
Decision Notification:	5th July 2009
Competition papers:	6th July 2009
Camera-ready:	31st July 2009
Symposium:	7-10 September 2009

Conference Venue

The symposium will be held at the Politecnico di Milano, the largest technical university in Italy, ten minutes from downtown Milan, the shopping area, and its famous galleries and museums.

For more information please visit: http://www.ieee-cig.org

July 2010

2010 IEEE World Congress on Computational Intelligence

July 18-23, 2010, Barcelona, Spain

Homepage: WWW

Deadline January 31, 2010

The 2010 IEEE World Congress on Computational Intelligence (IEEE WCCI 2010) is the largest technical event in the field of computational intelligence. It will host three conferences: the 2010 International Joint Conference on Neural Networks (IJCNN 2010), the 2010 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010), and the 2010 IEEE Congress on Evolutionary Computation (IEEE CEC 2010). IEEE WCCI 2010 will be held in Barcelona, a Mediterranean city located in a privileged position on the northeastern coast of Spain. Barcelona combines history, art, architecture, and charm within a pleasant, and efficient urban environment where meet old friends, and make new ones. The congress will provide a stimulating forum for scientists, engineers, educators, and students from all over the world to discuss and present their research findings on computational intelligence.

Important Due Dates

- Submission deadline: January 31, 2010
- Competition proposals: November 15, 2009
- Special sessions proposals: December 13, 2009
- Notification of special session acceptance: December 22, 2009
- Paper submission: January 31, 2010
- Tutorial and workshop proposal: February 14, 2010
- Notification of tutorial and workshop acceptance: February 22, 2010
- Notification of paper acceptance: March 15, 2010
- Final paper submission: May 2, 2010
- Early registration: May 23, 2010
- Tutorial and Workshops: July 18, 2010
- IEEE WCCI 2010 Conference: July 19, 2010

For more information visit http://www.wcci2010.org/call-for-papers

About the Newsletter

SIGEVOlution is the newsletter of SIGEVO, the ACM Special Interest Group on Genetic and Evolutionary Computation.

To join SIGEVO, please follow this link [WWW]

Contributing to SIGEVOlution

We solicit contributions in the following categories:

Art: Are you working with Evolutionary Art? We are always looking for nice evolutionary art for the cover page of the newsletter.

Short surveys and position papers: We invite short surveys and position papers in EC and EC related areas. We are also interested in applications of EC technologies that have solved interesting and important problems.

Software: Are you are a developer of an EC software and you wish to tell us about it? Then, send us a short summary or a short tutorial of your software.

Lost Gems: Did you read an interesting EC paper that, in your opinion, did not receive enough attention or should be rediscovered? Then send us a page about it.

Dissertations: We invite short summaries, around a page, of theses in EC-related areas that have been recently discussed and are available online.

Meetings Reports: Did you participate to an interesting EC-related event? Would you be willing to tell us about it? Then, send us a short summary, around half a page, about the event.

Forthcoming Events: If you have an EC event you wish to announce, this is the place.

News and Announcements: Is there anything you wish to announce? This is the place.

Letters: If you want to ask or to say something to SIGEVO members, please write us a letter!

Suggestions: If you have a suggestion about how to improve the newsletter, please send us an email.

Contributions will be reviewed by members of the newsletter board.

We accept contributions in LaTeX, MS Word, and plain text.

Enquiries about submissions and contributions can be emailed to editor@sigevolution.org.

All the issues of SIGEVOlution are also available online at www.sigevolution.org.

Notice to Contributing Authors to SIG Newsletters

By submitting your article for distribution in the Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library
- to allow users to copy and distribute the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.