*Genes, Genomes and Genomics* ©2008Global Science Books

# Computational Genome Analysis Using The G-language System

**Kazuharu Arakawa[§*] • Haruo Suzuki[§] • Masaru Tomita**

[1] Institute for Advanced Biosciences, Keio University, Fujisawa 252-8520, Japan
[§]First two authors are equal contributors
*Corresponding author*: * gaou@sfc.keio.ac.jp

## ABSTRACT

Computational genome analysis requires sophisticated workflows, seamlessly uniting multiple tools and algorithms. In order to maximize the productivity of genomics research with bioinformatics, a computational framework that allows rapid integration of available resources is desirable. G-language Genome Analysis Environment is a generic open-source workbench for this purpose, with the aim to: 1) construct an integrated analysis and development environment for bioinformatics, 2) systematically accumulate and implement existing algorithms and data, and to 3) aid the construction of analysis workflows. This system provides over 200 analysis methods for genome informatics and systems biology, with programmable interfaces, an interactive command-line shell, and a graphical user interface. Here we review the methods and algorithms implemented in this system especially focusing on genome informatics analysis, including methods for the identification of sequences with significant information content using information theory, observation of nucleotide composition and genomic compositional asymmetry, calculation of codon bias measures and prediction of gene expression levels, and statistical analysis of short oligomers such as short tandem repeats and palindromes. Since these methods are combined with several other applications and algorithms to produce a workflow in genome informatics research for studying specific biological questions, we also present brief overviews of workflows utilizing these algorithms used in several genomic studies.

**Keywords:** bioinformatics, codon, G-language Genome Analysis Environment, genome informatics, GC skew, genomics
**Abbreviation: PWM**, position weight matrix

## CONTENTS



## INTRODUCTION

Molecular biology has quickly become a data-driven science. Bioinformatics is now an indispensable means in order to cope with the 'explosion' of data, exemplified by the hundreds of completely sequenced genomes and evermore-increasing omics information produced by high-throughput experiments (Butler 2001; Arakawa *et al.* 2006b; Liolios *et al.* 2006). Computational biology has initially evolved synergistically with the genome projects, especially in sequence assembly, gene identification, and annotation of the genomic information, greatly contributing to the success of the Human Genome Project (Stein 1996; Chicurel 2002; Collins *et al.* 2003; Hood *et al.* 2003). At this stage the primary role of bioinformatics was data processing and generation, but with the progress in genomics and the advent of systems biology, it is rapidly expanding into the fields of knowledge discovery through data integration and mining of the masses of information, and hypothesis generation and testing (Kitano 2002a, 2002b; Kell *et al.* 2004; Ideker *et al.* 2006). Central to the advent of this new paradigm is the availability of effective software infrastructures. A myriad of bioinformatics tools have been developed for specific analyses, including *de facto* standard sequence analysis

Received: 10 January, 2008 Accepted: 26 July, 2008

Invited Review

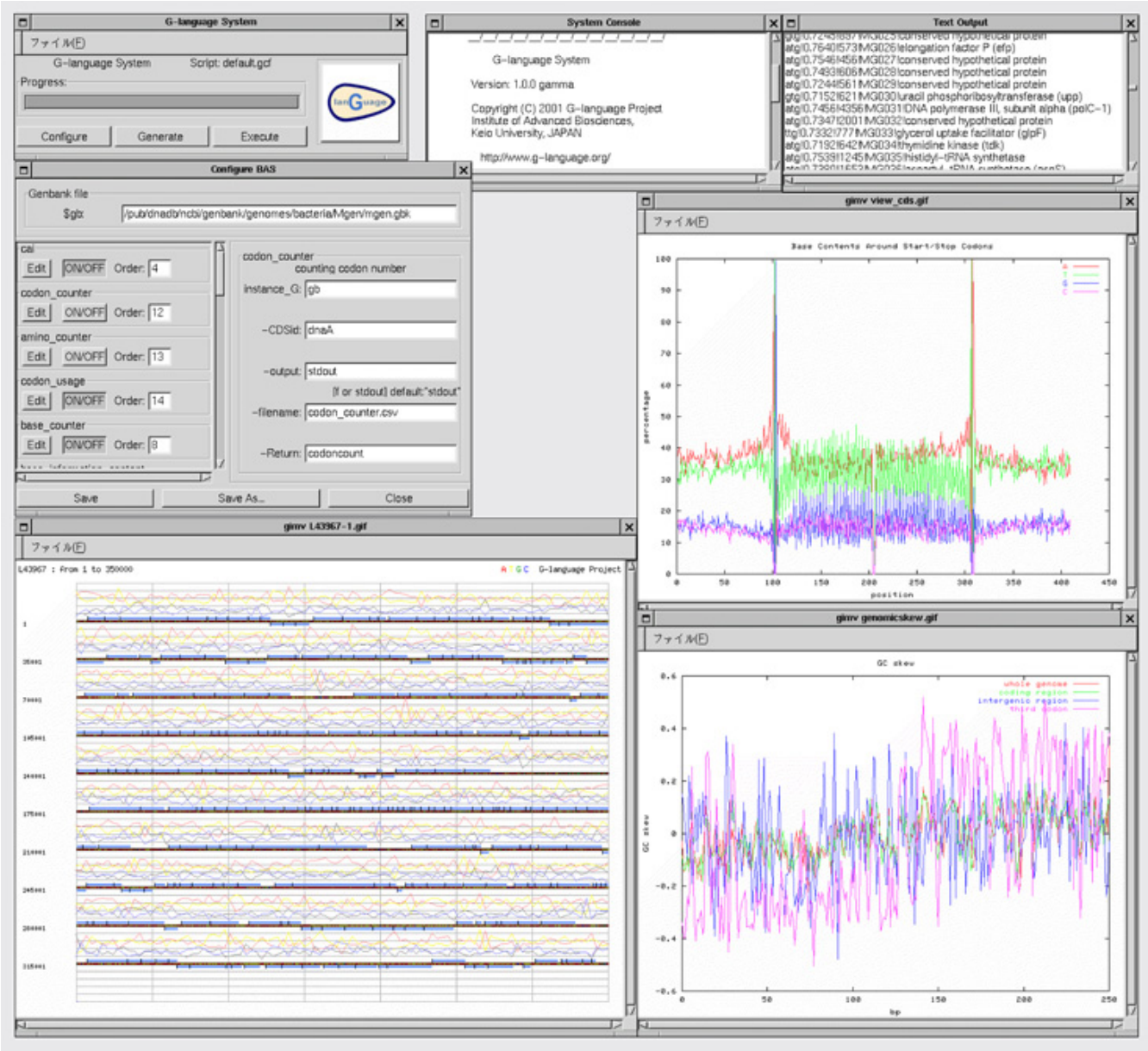


**Fig. 1 Graphical User Interface (GUI) of G-language GAE.** Viewing clockwise from top left corner, shown windows are 1) main control panel, 2) console, 3) text output window, 4) result from view_cds program showing nucleotide contents around start/stop codons, 5) result from genomicskew program showing GC skew of multiple regions of genome, 6) result from genome_map program showing gene locations and nucleotide contents, and 7) configuration window for the manipulation of workflow. Using the GUI, users can run the programs implemented in G-language GAE in a workflow without writing a single line of code.

software tools such as BLAST (Altschul *et al.* 1997), HMMER (Eddy 1998), and ClustalW (Thompson *et al.* 1994).

Since computational biology encompasses extremely broad areas of molecular biology that are at the same time rapidly expanding and evolving, development of any one feature-rich and versatile application for bioinformatics would likely not work to suit the diversity of research fields. Instead, specialized software tools and components that are developed to perform specific analyses can be dynamically linked together, creating a workflow to achieve the intended research depth (Swertz *et al.* 2007). Development of tailor-made software by combining functional units to meet diverse needs using the command-line environment of UNIX operating systems has been shown to be efficient, because the programs can be "piped" to perform complex functions. For example, the combination of merely 10 command-line programs is possible in theory to generate 10! = 3,628,800 variations, which requires much less development cost than one multi-functional application with 100 features. A similar design principle can be seen in the organization of the human genome, where extremely diverse biological functions arise from a limited number of genes. This approach, called "mash-up", is actively utilized in the current developments of web-applications as a cost-effective means to create diverse products (Belleau *et al.* 2008). In order to connect different software tools in a pipeline, at lease some computer programming is essential to filter the inputs and outputs, and a software infrastructure that allows minimal programming and easy connection of the components is necessary for effective computational molecular biology.

Current efforts for such bioinformatics software infrastructures mainly focus on one of the three interfaces of the computational environment: application programming interface (API), command-line user interface (CUI), and graphical user interface (GUI). BioPerl, BioPython, BioJava, and BioRuby, collectively known as the Bio* toolkits (Mangalam 2002; Stajich *et al.* 2002), provide APIs for easy handling of the various biological databases and software tools in corresponding programming languages. With these toolkits, bioinformatics developers can access the data as native objects without worrying about the differences in data formats, and at the same time take advantage of the text

string processing abilities of the scripting languages that are especially convenient for biological sequence information. The Bioconductor project (Gentleman *et al.* 2004) also provides APIs for bioinformatics especially focusing on gene expression analyses, data visualization, and statistical analyses, taking advantage of the rich features of R statistics language, which is available at http://www.r-project.org/. Since the R language is equipped with an interactive shell environment, the user experience of Bioconductor is similar to that of CUI. EMBOSS (European Molecular Biology Open Source Software Suite) (Rice *et al.* 2000) and NCBI SEALS (System for Easy Analysis of Lots of Sequences) (Walker *et al.* 1997) provide large collections of command-line applications for data retrieval and manipulation, sequence analyses, phylogenetic analyses, and numerous others that can be linked into workflows. The majority of bioinformatics software is distributed in the form of command-line applications, and therefore these toolkits can work seamlessly together. Moreover, these command-line tools are also often provided as web-services based on Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) with BioMOBY standards (Wilkinson *et al.* 2005), and Taverna workbench provides a GUI to utilize these web-services and to formulate workflows (Oinn *et al.* 2004). It is worth noting that these projects are mostly based on open-source development for high accessibility, dynamism, and transparency that is required for scientific purposes in order to be able to examine the methods and algorithms employed within the software tools. All of these interfaces are practical solutions depending on the use cases, due to the tradeoffs between scalability and accessibility. Programming using APIs is the most scalable interface, but it is also the least accessible one since it requires certain level of programming knowledge and skills. Conversely, GUI is most accessible with the intuitive interface that can be manipulated with a mouse, but applications of this type are limited in terms of scalability. Multiple interfaces are sometimes employed to improve the flexibility of the software. For example, EMBOSS also provides comprehensive development APIs and a graphical front-end named Jemboss (Carver *et al.* 2003), and BioPerl includes interfaces for EMBOSS. SOAP-based web-services can be accessed through APIs, and Taverna also allows scripting by Java within the GUI application for customized filtering of the data.

Bioinformatics currently belongs to the molecular biology domain, and therefore genome sequence data is central to this discipline. The digital nature of biological sequence information makes computer programming a highly suitable means for the manipulation of this data (Hood *et al.* 2003), especially for scripting languages such as Perl that is specialized for text processing. Programming is also necessary for genome informatics research in terms of scalability of software development by mash-ups of existing tools and algorithms. However, interactive and responsive CUI as well as accessible and user-friendly GUI would also facilitate the research processes. In light of these requirements for computational genome informatics, the G-language Project at the Institute for Advanced Biosciences, Keio University, Japan have been developing a generic workbench designated G-language Genome Analysis Environment (G-language GAE) since 2001 (Arakawa *et al.* 2003). The software system is equipped with interchangeable interfaces as API, CUI, and GUI for high scalability and accessibility, with Perl-based API that is compatible with BioPerl, more than 200 analysis applications especially focusing on genome informatics studies many of which are not available through other software packages, and an intuitive GUI that can be easily converted from Perl scripts (**Fig. 1**). G-language GAE is developed as an open-source software, distributed under GNU General Public License at http://www.g-language.org/. Currently available version of the software is 1.8.4, but version 2 with enhanced user interface is also under development. Details of the internal architecture of the software system is reviewed elsewhere (Arakawa *et al.* 2006a).

G-language GAE is especially strong for genome informatics analyses, with numerous algorithms implemented to be directly accessible from Perl programming language (and with BioPerl sequence object). In this work, we review the methods and algorithms implemented in G-language GAE for genome informatics studies, concentrating on the following areas: identification of binding sites based on information theory, analysis of genomic compositional skew, and analysis of synonymous codon usage bias. Combined with other software tools and scripts, utilization of these algorithms should facilitate the development of workflows for computational genome analysis.

## IDENTIFICATION OF BINDING SITES WITH INFORMATION THEORY

A typical starting point in sequence analysis is the identification of conserved sequence elements or motifs in order to characterize functional sequence structures, such as transcription factor binding sites and Shine-Dalgarno (Shine *et al.* 1974)/Kozak (Kozak 1987) sequences for ribosome binding sites. In order to identify such binding sites, multiple sequences are typically aligned to formulate a position weight matrix (PWM) (Stormo *et al.* 1982), and conservation is thereby quantified mathematically based on Claude Shannon's information theory (Shannon 1948), which is applied to nucleotide and protein sequences by the works of Schneider and colleagues (Schneider 1997; Schneider 2002). Commonly used indices for this purpose include Shannon uncertainty (entropy) $H$, information content $I$, and Kullback-Leibler divergence. Sequence Logo software (Schneider *et al.* 1990) and WebLogo online generator (Crooks *et al.* 2004) are frequently utilized to visualize the information content in a given set of aligned sequence data (**Fig. 2**). Naïve calling of the most frequent alphabets for the identification of "consensus" sequence can be erroneous, since such method ignores the frequency information and subsequently the degree of sequence conservation in the given alignment. Methods based on information theory avoid this pitfall by measuring the amount of information (or randomness) to identify conserved residues (Schneider 2002).

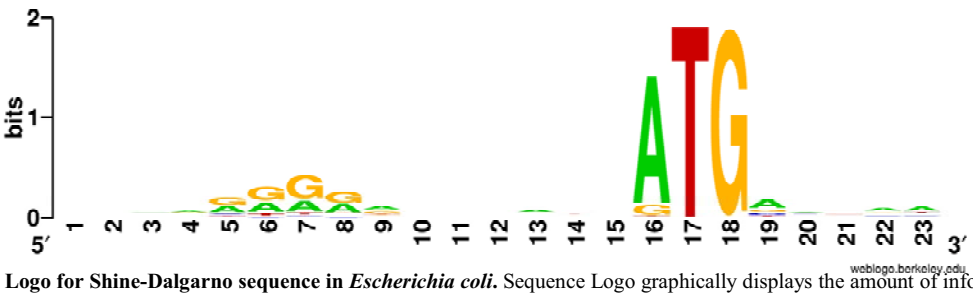


**Fig. 2 Sequence Logo for Shine-Dalgarno sequence in *Escherichia coli*.** Sequence Logo graphically displays the amount of information content at each position, represented by the height of the stacked alphabet. Height of each nucleotide corresponds to its contributing frequency. Here one can clearly see the conservation of ATG start codon, and purine rich Shine-Dalgarno sequence can be found at positions -7 to -12 (4 to 9 in the figure). WebLogo (Crooks *et al.* 2004) was utilized for visualization.

**Table 1** Programs for consensus analysis implemented in G-language GAE.

| Name | Description |
|---|---|
| base_entropy | Calculates and graphs the sequence conservation in regions around the start/stop codons using Shanon uncertainty (entropy). |
| base_information_content | Calculates and graphs the sequence conservation in regions around the start/stop codons using information content. |
| base_relative_entropy | Calculates and graphs the sequence conservation in regions around the start/stop codons using Kullback-Leibler divergence (relative entropy). |
| consensus_z | Calculates and graphs the sequence conservation in a given array of sequences, and names a consensus using *z*-score cutoff. |

When conserved regions are identified with information theory, PWM for the putative binding site of interest can be used as training set for machine learning methods, for further search of similar conserved regions and motif prediction (Cartharius *et al.* 2005; Hertzberg *et al.* 2005). HMMER software based on Hidden Markov Model (HMM) algorithm has been especially popular in genomics and proteomics (Eddy 1998). Other machine learning approaches frequently utilized for the prediction of protein binding sites include artificial neural networks, self-organizing maps, and support vector machines (SVM), that are reviewed elsewhere (Schneider *et al.* 2004).

In the following paragraphs, we describe three methods based on the information theory for the analysis of sequence conservation: uncertainty, information content, and Kullback-Leibler divergence. In addition, we describe the *z*-score method for consensus sequence calling by statistical means. All of these methods are implemented in G-language GAE (**Table 1**).

### Shannon uncertainty and information content

Uncertainty of information $H$ (also known as entropy) at given position $i$ with distribution $P_i$ is defined as follows:

$$H(P_i) = -\sum_{j \in M} P_{ij} \log_2 P_{ij} \qquad \text{(bits per symbol)}$$

where $M$ is the set of alphabets representing the sequence units (for DNA, {A, T, G, C}) and $P_{ij}$ is the frequency of a certain alphabet at position $i$ (Shannon 1948). Unit of $H$ is "bits" when binary logarithm is used, as in the above formula. $H(P_i)$ is zero and minimum when distribution is most biased and therefore the entire information is represented by only one letter of the alphabet. $H(P_i)$ takes the maximum value of $\log_2 |M|$ bits when the distribution is uniform for all alphabets, where $|M|$ is the cardinality of $M$ (4 for DNA, therefore the maximum $H$ is 2).

Information content $I$ is obtained by subtracting $H$ from the maximum uncertainty $\log_2|M|$,

$$I(P_i) = \log_2 |M| - H(P_i) = \log_2 |M| - \left( -\sum_{j \in \Sigma} P_{ij} \log_2 P_{ij} \right)$$

therefore $I(P_i)$ is maximum when the frequency is most biased to certain single alphabet (Schneider *et al.* 1990).

### Kullback-Leibler divergence (relative entropy)

Above described uncertainty and information content assume uniform distribution of all alphabets. However, in realistic conditions, background nucleotide composition (e.g., genomic G+C content) varies among species, and distribution is even more diverse for protein sequences. In order to account for this heterogeneity of the innate distributions of the alphabets, relative entropy $H(P_i \parallel \pi)$ or Kullback-Leibler divergence from given background distribution $\pi$ is derived as follows:

$$H(P_i \| \pi) = H(P_i, \pi) - H(P_i) = \sum_{j \in M} P_{ij} (\log_2 P_{ij} - \log_2 \pi_j)$$

$$= \sum_{j \in M} P_{ij} (\log_2 \frac{P_{ij}}{\pi_j})$$

where $\pi_j$ is the background frequency of alphabet $j$. Note that $H(P_i \parallel \pi) = I(P_i)$ when $\pi = 1/|M|$. Sequence Logo for protein motifs is developed using Kullback-Leibler divergence by Schuster-Böckler *et al.* (2004).

Although Kullback-Leibler divergence is commonly utilized to quantify the conservation in amino acid sequences, one should note that this is a divergence measure from the background distribution and it is not an information measure (Schneider 1999). This measure is called "divergence" since the calculation is asymmetric and therefore not sufficient as a distance measure. Moreover, Kullback-Leibler divergence can result in values exceeding the maximum uncertainty $\log_2|M|$, and consequently it is inappropriate to be considered using the unit of "bits".

### *z*-score cutoff

Simplest means for the statistical testing of the significance of conservation is the use of *z*-score (also called the standard score), defined as:

$$z = \frac{P_{ij} - \mu}{\sigma}$$

where $\mu$ is the mean and $\sigma$ is the standard deviation of all $P_{ij}$, for the most frequent alphabet $j$. The *z*-score represents how many standard deviations a raw score deviates from the mean. For example, representation of most frequent alphabet scoring $z > 2$ with upper case letters and those $2 > z > 1$ with lower case letters for regions surrounding the start codon in *Escherichia coli* K12 shown in **Fig. 2** results in a following putative binding site: --a-ggGga---a--ATGaa-aa. Note that the *z*-score statistics can be used to identify significant positions within a binding site, but this is different from the amount of sequence conservation computed by the information theory. Comparison of indices based on information theory with *z*-score cut-off is discussed in a work using human cDNA sequences and G-language GAE (Arakawa *et al.* 2005b).

## ANALYSIS OF NUCLEOTIDE COMPOSITION BIAS

A genome is primarily shaped by the requirements of its coded genes, but at the same time, it is also highly organized as a functional medium that undergo replication, especially in fast growing bacteria where doubling time is in the order of less than a couple of hours. Circular bacterial chromosomes have single finite origin of replication from which replication forks progress bi-directionally, until the two forks meet at the replication terminus typically located directly opposite of the origin, maintaining a physical balance (Rocha 2004b, 2004c). Therefore, a single strand of circular bacterial chromosome is divided into two replichores by the replication origin and terminus, where the two replichores correspond to the leading strand of one replication arm and the lagging strand of another arm, respectively (Lobry *et al.* 2003). Because of the discontinuous strand synthesis in the lagging strand, mutational bias in the two strands of DNA molecule result in the asymmetry in nucleotide composition (Lobry 1996; Frank *et al.* 1999; Lobry *et al.* 2002). This characteristic genomic polarity can be visualized by plotting the relative abundance of C over G along the genomic positions, which is known as the GC skew graph (*vide infra*). FtsK translocase senses this genomic

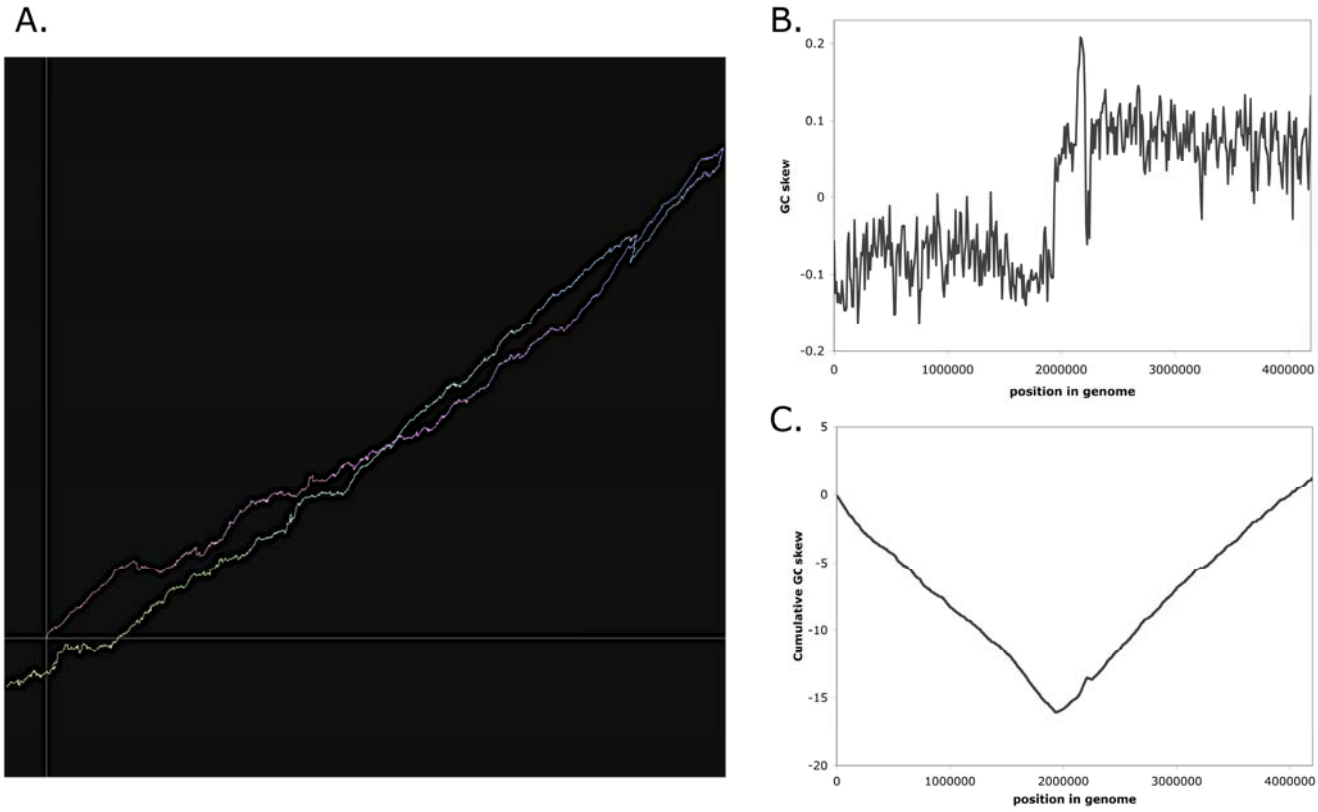


**Fig. 3 Compositional asymmetry of *Bacillus subtilis* genome.** (**A**) DNA walk graph: starting from the origin (where thin white axis cross), pixel is moved and colored according to the type of the nucleotide (A-up, T-down, G-right, C-left). (**B**) GC skew graph: G-rich region shifts to C-rich region near the replication terminus located near 2,000,000 bp position. (**C**) Cumulative GC skew graph: shift point of GC skew becomes clearly visible as the maxima and minimum.

compositional asymmetry by recognizing KOPS oligomers in *E. coli*, in order to identify the *dif* sequence located near the replication terminus targeted by XerCD during chromosome dimer resolution (Perals *et al.* 2000, 2001; Levy *et al.* 2005; Pease *et al.* 2005; Bigot *et al.* 2007). Accordingly, regions surrounding the replication terminus are enriched in skewed oligomers (Hendrickson *et al.* 2006), and coupled with the A+T enrichment around the replication terminus, background nucleotide composition bias affect the codon usage of genes depending on their locations within the genome (Daubin *et al.* 2003).

Near the replication origin, genes are preferentially located in the leading strand, especially for longer genes, operons, highly expressed genes, and essential genes (McLean *et al.* 1998; Rocha *et al.* 2003 Omont *et al.* 2004; Price *et al.* 2005). These tendencies are speculated to be formed by avoiding head-on collisions of DNA and RNA polymerases in the interplay of replication of transcription (Brewer 1988; Liu *et al.* 1995), and also due to the gene dosage effects of fast growing bacteria where multiple rounds of replication take place, especially for genes related to transcription and translation (Ardell *et al.* 2005; Couturier *et al.* 2006). In light of the many replication-related constraints that govern the chromosomal organization in bacterial genomes, it is essential to have the knowledge of accurate positions of replication origin and terminus, and subsequently the sequences for leading and lagging strands. Computational prediction of replication origin is a common practice in genome projects as a cost effective and sufficiently accurate alternative to experimental methods, and most bacterial genome projects use putative replication origin as the first base position when submitting the sequence data to public repositories.

## GC skew analysis

GC skew is defined as the excess of C over G normalized by the G+C content [(C – G) / (C + G)] in a given region (Lobry 1996). By graphing the GC skew values continuously along the genome sequence using sliding windows (for example, 10000 bp), most bacterial genomes are divided into C-rich lagging strand and G-rich leading strand (**Fig. 3B**). Replication origin and terminus are located in the vicinity of the shift points between two strands. Cumulative graph of GC skew is a frequently utilized alternative to clarify the shift points, where the maximum and minimum points correspond to the replication origin and terminus, respectively (Grigoriev 1998) (**Fig. 3C**). GC skew can be observed in both coding and intergenic regions, and sometimes GC skew of third codon positions is preferred to avoid the compositional bias of the coding regions (Frank *et al.* 2000). Similarly derived AT skew [(A – T) / (A + T)] shows less significant polarity compared to GC skew, but the use of keto excess (G + T – C – A) / (A + T + G + C) or purine excess (G + A – C – T) / (A + T + G + C) is suggested to be more accurate in prediction of the replication origin and terminus for some bacterial species (Freeman *et al.* 1998). All of these measures are partial projections of the DNA walk diagrams, a pseudo-random walk representation of all nucleotides in a sequence, which is the trail drawn by moving a pixel in the direction of the type of nucleotide (in **Fig. 3A**, A-up, T-down, G-right, C-left).

## Prediction of replication origin and terminus

Sequence-based prediction of replication origin and terminus identifies the peak positions of the aforementioned cumulative skew graphs. Oriloc is a popular implementation of this kind of algorithm, which detects the intersection of the DNA walk trajectory and its linear regression based on the nucleotide content of third codon positions (Frank *et al.* 2000). Although these methods are sufficiently accurate especially for replication origin, prediction of replication terminus usually has an error margin of around 10 kbp from the experimentally identified sites, due to high insertion and horizontal transfer rates (Moszer *et al.* 1991). To improve

**Table 2** Programs for genomic compositional analysis and sequence pattern searches implemented in G-language GAE.

| Name | Description |
|---|---|
| gcskew | Calculates and graphs the GC skew of the genome. By specifying the optional parameter, this method can also graph AT skew, purine and keto excess, and cumulative skew. |
| genomicskew | Graphs the GC skew of whole genome, coding regions, GC3, and intergenic regions. Optionally shows AT skew or purine / keto excess for those regions. |
| dnawalk | Graphs the DNA walk of given sequence. |
| find_ori_ter | Predicts the replication origin and terminus by identifying the peaks of cumulative GC skew graph at single base-pair resolution. Optionally uses AT skew or purine / keto excess for prediction, and the use of FFT-based noise reduction filtering can be specified. |
| rep_ori_ter | Returns the locations of replication origin and terminus by referring to the internal database of experimentally verified loci. If data is not available in internal database, loci are predicted using find_ori_ter. |
| gcsi | Quantifies the degree of skew (GCSI). |
| leading_strand | Returns the sequence of leading strand. |
| query_strand | Given a position and direction of the strand, returns whether it is on the leading or lagging strand. |
| query_arm | Given a position, returns whether it is on the left or right arm of replication. |
| set_strand | Sets the strand information for all genes. |
| set_gc3 | Sets the GC3 information for all genes. |
| genes_from_ori | Retrieves gene names in the order of distance from the origin. |
| dist_in_cc | Calculates the distance of given position from the origin. |
| gcwin | Calculates and graphs the GC content along the chromosome. Optionally calculates AT content. |
| signature | Calculates the oligonucleotide relative abundance (genomic signature). |
| palindrome | Searches for palindrome sequences of given length. |
| find_dif | Searches for *dif* sequence |
| find_ter | Searches for Ter sites |
| find_dnaAbox | Searches for dnaA-boxes |
| find_iteron | Searches for Iterons |
| oligomer_search | Searches for given oligomer. Degenerate nucleotide code or regular expressions can be used. |

the prediction accuracy for terminus regions and to clarify the skew shift points under the presence of background "noise", low-pass filtering using Fast Fourier Transform (FFT), a common method to reduce innate noise in image or signal processing disciplines, has been proven to be successful (Arakawa *et al.* 2007a). In low-pass filtering, a given discrete signal with length $N$, $f$(n), n = 0, 1, …, $N$ – 1, at frequency $k$, is transformed to frequency domain representation by FFT as follows,

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-i2\pi kn/N}$$

where $i = \sqrt{-1}$. The power spectrum $PS(k)$ of $F(k)$ is subsequently obtained as follows,

$$PS(k) = |F(k)|^2, k = 0,1,2,...,N-1$$

at each frequency $k$. Since noise in data are distributed in the high frequency domain and replication-related selection should belong in low frequency domain considering the global nature of replication, zero-ing the power spectrum of the high frequency regions (thus "low-pass") and reverse transforming the spectrum regenerates a skew graph with reduced noise. Obtaining the peak positions of the noise-filtered cumulative skew graph results in better prediction.

In addition to the skew-based methods, locations of *cis*-acting sequence elements related to replication are typically combined to support the computational prediction. For example, DnaA boxes (5′-TTATCCACA-3′ in *E. coli*) where DnaA proteins bind to unwind the DNA molecule in order to initiate the replication fork are indicative of the position of replication origin (Kaguni 1997), and the orientation of Ter sites where Tus proteins bind to block the replication fork progression only in one direction (Hill 1992) as well as the previously described *dif* sequence help identify the replication terminus. In bacterial plasmids, iteron sequences (5′-TGAGGG G/A C/T-3′) are indicative of replication origins (Haines *et al.* 2006). All methods described in this section including the identification of these sequence features are implemented in G-language GAE (See **Table 2** for complete listing).

## GC Skew Index

Although GC skew is commonly observed in a wide variety of bacterial species, the "degree" or "visibility" of the skew is quite diverse. For example, slow growing bacteria such as Cyanobacteria and Mycoplasma exhibit only weak skews, and archaea do not show visible skew due to their different replication machinery (Daubin *et al.* 2003). Suitability of GC skew-based prediction methods for replication origin and terminus significantly relies on the clarity of GC skew, and therefore it is useful to quantify the degree of skew to allow comparative studies. GC Skew Index (GCSI) is a quantitative indicator for this purpose, calculated by combining the spectrum ratio (*SR*) between 1Hz spectrum and the average of all spectra of 2Hz and above of FFT, and the Euclidean distance (*dist*) between the two vertices of cumulative graph. Here *SR* captures the fitness of the "shape" of GC skew graphs, and *dist* measures the degree of bias (Arakawa *et al.* 2007c). GCSI is normalized to range from 0 to 1, and genomes with GCSI < 0.05 have no observable skew (majority of archaeal genomes belong to this category). *E. coli* has GCSI of around 0.1, above which clear shift points can be discerned. Therefore, GCSI should be a useful criterion to test the applicability of skew-based predictions, although one should also note that GCSI is not necessarily a measure of replication selection (Arakawa *et al.* 2007b).

## G+C content

G+C content is the percent of guanine and cytosine in the nucleotide sequence, expressed as 100 x (G + C)/(A + T + G + C). Genomic G+C content varies widely among different bacterial species, and this variation is most pronounced at the third position of codons because the first two positions of codons are constrained by protein-coding requirements (Muto *et al.* 1987). For example, among 80 bacterial species tested by Sharp *et al.* (2005), genomic G+C content ranged from 22 to 72%, whereas G+C content at synonymously variable third positions ranged from 9 to 93%. Various factors have been proposed as determinants of G+C content, including genome-wide mutational bias toward G+C or A+T (Sueoka 1962), higher energy cost and limited availability of G+C over A+T (Rocha *et al.* 2002), increment in G+C in aerobiosis (Naya *et al.* 2002), and

horizontal DNA transfer among distantly related species with different genomic G+C contents (Lawrence *et al.* 1997). To identify putative foreign genes, G+C content is determined for overall protein-coding regions and/or at different positions of codons (Lawrence *et al.* 1997 Garcia-Vallve *et al.* 2000). To identify genomic islands (clusters of foreign genes), G+C content is computed using sliding windows (Karlin 2001), and this is sometimes also useful to identify the putative coding regions especially in A+T rich genomes (Chen *et al.* 2004b).

### Genomic signature

Karlin and his coworkers proposed that each organism has its characteristic "genomic signature" defined as the ratios between the observed and expected frequencies of dinucleotides (dinucleotide relative abundances) (Karlin *et al.* 1995; Karlin *et al.* 1998a). The dinucleotide relative abundance value ($\rho^*_{XY}$) is calculated as:

$$\rho^*_{XY} = f^*_{XY} \Big/ f^*_X f^*_Y$$

where $f^*_X$ and $f^*_Y$ denote the frequency of the mononucleotide $X$ and $Y$ respectively, and $f^*_{XY}$ denotes the frequency of the dinucleotide $XY$, computed from the sequence concatenated with its inverted complement sequence. Because the genomic signature is relatively constant throughout the genome and similar between closely related species, it has been used to construct phylogenetic trees (Coenye *et al.* 2003; Coenye *et al.* 2004; van Passel *et al.* 2006) and to detect anomalous genomic regions such as genomic islands (Karlin 2001; van Passel *et al.* 2005). The analysis of different word length (e.g., 4-letter words = tetranucleotide) in different size of sliding window (Dufraigne *et al.* 2005) can be implemented in G-language GAE.

## ANALYSIS OF SYNONYMOUS CODON USAGE BIAS

Much genetic code is degenerate, meaning that most amino acids are encoded by more than one codon (triplet of nucleotides); these synonymous codons usually differ by one nucleotide in the third position. Synonymous codons are not used with equal frequency, and their usage varies among different species and also among genes within the same genome (Sharp *et al.* 1988). Different factors have been proposed to explain variations in synonymous codon usage among genes, including genome-wide mutational bias (shaping intergenomic variation in G+C content) (Chen *et al.* 2004a), natural selection linked to optimal growth temperature (Lynn *et al.* 2002; Lobry *et al.* 2006), horizontal gene transfer among distantly related species (shaping intragenomic variation in G+C content) (Lawrence *et al.* 1997; Garcia-Vallve *et al.* 2000), strand-specific mutational bias (shaping GC skew between leading and lagging strands of DNA replication) (McInerney 1998; Lafay *et al.* 1999), and natural selection for translation optimization (acting mainly on highly expressed genes) (Ikemura 1985; Rocha 2004a; Sharp *et al.* 2005). There are many statistical methods to analyze synonymous codon usage bias (Comeron *et al.* 1998; Ermolaeva 2001). G-language GAE is more comprehensive for the analysis of synonymous codon usage bias, compared to the most popular package such as CodonW (available at http://codonw.sourceforge.net/). The analysis methods include (i) normalization of codon usage data, (ii) multivariate analysis of codon usage data, (iii) measure of synonymous codon usage evenness, and (iv) prediction of gene expression level from codon usage. We will discuss the performance and instruction for the use of these methods (**Table 3**).

**Table 3** Programs for codon analysis implemented in G-language GAE.

| Name | Description |
|---|---|
| codon_usage | Displays the codon table of the given genome or specified gene. |
| codon_mva | Performs multivariate analyses of codon usage data, and analyzes correlations between the axes and other gene features such as G+C content and GC skew. |
| enc | Calculates the effective number of codons (*Nc*). |
| cbi | Calculates the codon bias index (CBI). |
| icdi | Calculates the intrinsic codon deviation index (ICDI). |
| Ew | Calculates the weighted sum of relative entropy (*Ew*). |
| P2 | Calculates the P2 index. |
| fop | Calculates the frequency of optimal codons (*Fop*). |
| w_value | Calculates the 'relative adaptiveness (W) of each codon. |
| cai | Calculates codon adaptation index (CAI) for each gene. |
| phx | Calculates the expression measure, *E*(*g*), to identify predicted highly expressed (PHX) genes. |

### Representation of codon usage data

Five different kinds of representations of codon usage data (termed here $R_0$-$R_4$) have been used in codon usage studies (Perriere *et al.* 2002; Suzuki *et al.* 2005). For a single gene or a group of genes, the value of the *j*th codon for the *i*th amino acid ($x_{ij}$) is defined as:

For $R_0$, $x_{ij} = n_{ij}$

For $R_1$, $x_{ij} = \dfrac{n_{ij}}{\sum_{i=1}^{20}\sum_{j=1}^{k_i} n_{ij}}$

For $R_2$, $x_{ij} = \dfrac{n_{ij}}{\sum_{j=1}^{k_i} n_{ij}}$

For $R_3$, $x_{ij} = \dfrac{n_{ij}}{\frac{1}{k_i}\sum_{j=1}^{k_i} n_{ij}}$

For $R_4$, $x_{ij} = \dfrac{n_{ij}}{n_{i\max}}$

where $n_{ij}$ is the number of *j*th codon for the *i*th amino acid, $k_i$ is the degree of codon degeneracy for the *i*th amino acid (e.g., $k_i = 2$ for cysteine and $k_i = 6$ for arginine), and $n_{i\max}$ is the number of the most frequently used synonymous codon for the *i*th amino acid. The codon usage data $R_0$, $R_2$, $R_3$, and $R_4$ are also called as the absolute codon frequency (AF), the relative codon frequency (RF), the relative synonymous codon usage (RSCU), and the relative adaptiveness (W) of each codon, respectively (Sharp *et al.* 1986, 1987; Perriere *et al.* 2002).

### Multivariate analyses of codon usage data

Multivariate analysis methods, such as correspondence analysis (Grantham *et al.* 1980) and principal component analysis (PCA) (Kanaya *et al.* 1996), are often used to identify gene features contributing to the variations in synonymous codon usage among genes. Different kinds of codon usage data have been applied to these multivariate analysis methods (Perriere *et al.* 2002; Suzuki *et al.* 2005). Of the five codon usage data ($R_0$-$R_4$), only $R_4$ is independent of all three biases: (i) gene length, (ii) amino acid composition, and (iii) codon degeneracy. Indeed, PCA of $R_4$ data (PCA-$R_4$) is not affected by any of these biases (Suzuki *et al.* 2005). Consequently, PCA-$R_4$ is more effective than the other four methods at detecting gene features related to synonymous codon usage variations such as G+C content (G + C)/(A + T + G + C) at the third codon position (GC3) and GC skew (C - G)/(C - G) at the third codon position (**Fig. 4**).

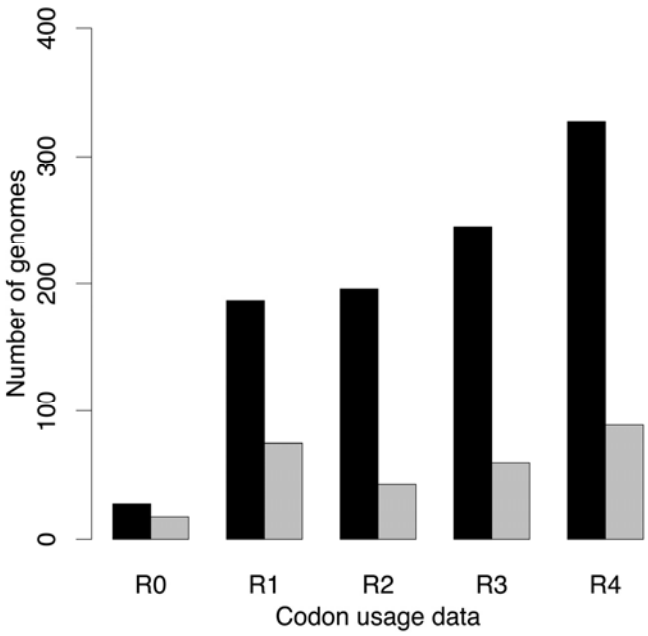


**Fig. 4 Performance comparison of principal component analysis (PCA) of different codon usage data ($R_0$, $R_1$, $R_2$, $R_3$, and $R_4$).** The performance was evaluated by the number of genomes where the gene feature – G+C content (black) and GC skew (gray) at third codon position – was detected on one of four axes generated by PCA in 559 bacterial genomes. The gene feature was detected when its correlation coefficient with the axis was the highest and greater than 0.7.

**Fig. 5** shows the plots of first and second axis scores obtained by PCA-$R_4$ for all individual genes in two fast growing bacteria *E. coli* and *Bacillus subtilis* as examples. The distribution of points reveals two horns: that corresponding to constitutively highly expressed genes (encoding ribosomal proteins and elongation factors) and that corresponding to putative foreign genes (Medigue *et al.* 1991; Moszer *et al.* 1999).

## Measure of synonymous codon usage evenness

Various measures of synonymous codon usage evenness have been proposed, including the 'effective number of codons' ($N_c$) (Wright 1990), the codon bias index (*CBI*) (Morton 1993), the intrinsic codon deviation index (*ICDI*) (Freire-Picos *et al.* 1994), Shannon uncertainty (entropy) from information theory ($H_s$) (Zeeberg 2002) and its modification, called the 'weighted sum of relative entropy' ($E_w$) (Suzuki *et al.* 2004). The entropy of the *i*th amino acid is defined as:

$$H_i = -\sum_{j=1}^{k_i} x_{ij} \log_2 x_{ij}$$

where $x_{ij}$ is the relative codon frequency ($R_2$) of the *j*th codon for the *i*th amino acid, and $k_i$ is the degree of codon degeneracy for the *i*th amino acid. $H_s$ and $E_w$ is calculated by combining the entropies from different amino acids:

$$H_s = \sum_{i=1}^{20} H_i$$

$$E_w = \sum_{i=1}^{20} \left( \frac{H_i}{\log_2 k_i} \right) p_i$$

where $p_i$ is the relative frequency of the *i*th amino acid in the protein. $E_w$ ranges from 0 (maximum bias) to 1 (maximum evenness). Because $E_w$ takes into account all three aspects of amino acid usage, i.e., (i) the number of different amino acids, (ii) their relative frequency, and (iii) their codon degeneracy, it is little affected by amino acid usage biases (Suzuki *et al.* 2004).

## Predicting gene expression level from codon usage

Various methods of predicting gene expression level from codon usage bias have been proposed, including the P2 index (Gouy *et al.* 1982), the frequency of optimal codons ($F_{OP}$) (Ikemura 1985), the codon adaptation index (CAI) (Sharp *et al.* 1987), and the expression measure, $E(g)$, for identifying predicted highly expressed (PHX) genes (Karlin *et al.* 2000). In some species, putative highly expressed genes (e.g., those encoding ribosomal proteins) do not have unusual codon usage, and thus codon usage cannot be used to predict gene expression levels (Grocock *et al.* 2002; Carbone *et al.* 2003). Therefore, to estimate the level of gene expression from codon usage bias, it is necessary first to check whether a genome shows evidence of translationally

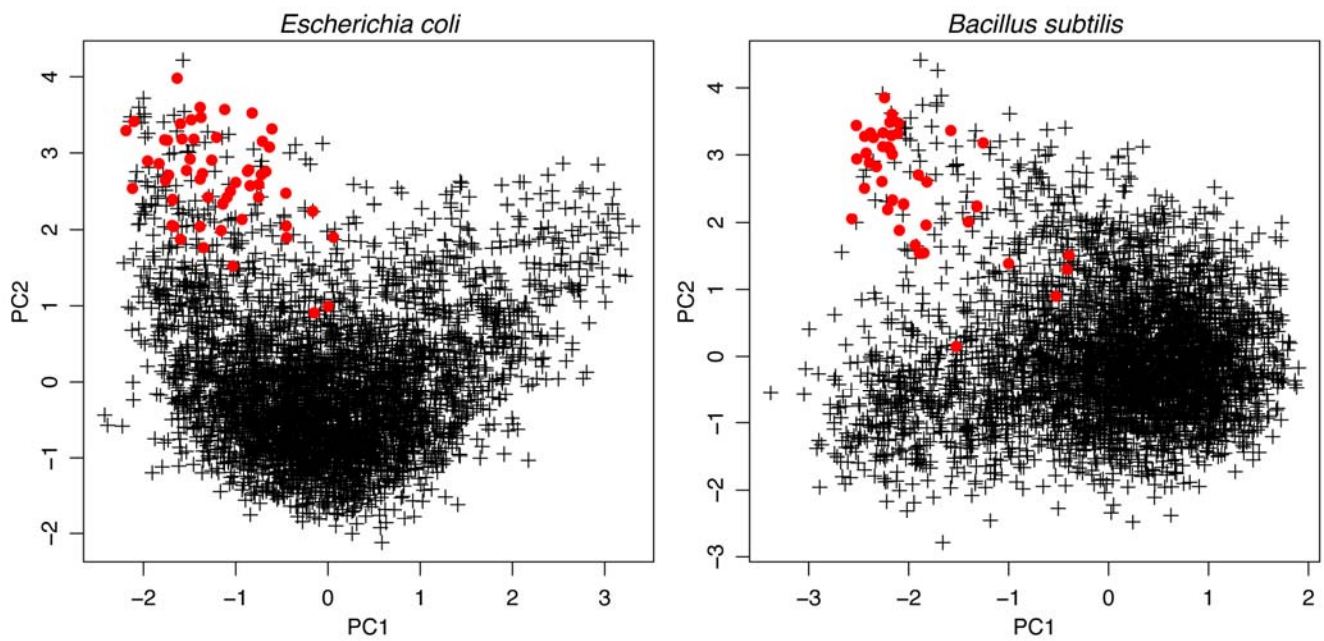


**Fig. 5 Plots of first and second axis (PC1 and PC2) scores obtained by PCA-$R_4$ for all individual genes in two fast growing bacteria *Escherichia coli* K12 and *Bacillus subtilis*.** Genes encoding ribosomal proteins are indicated by red circles.

selected codon usage bias by comparing codon usage of highly expressed genes with that of all genes (Henry *et al.* 2007), as shown in **Fig. 5**.

**P2 index.** The P2 index represents the proportion of codons conforming to the intermediate strength of codon-anticodon interaction energy rule of Grosjean and Fiers (Grosjean *et al.* 1982), and calculated as: P2 = (WWC + SSU) / (WWY + SSY), where W = A or U, S = G or C, and Y = C or U (Gouy *et al.* 1982). It indicates the efficiency of the codon–anticodon interaction and has been used as an indicator of the presence of translational selection (von Samson-Himmelstjerna *et al.* 2003). In fast growing bacteria such as *E. coli*, highly expressed genes have high P2 values (0.7-0.9), while other genes have values close to 0.5 (Shields *et al.* 1987).

**Frequency of optimal codons ($F_{OP}$).** In fast growing bacteria such as *E. coli*, highly expressed genes preferentially use optimal codons, which optimize the efficiency of translation (determined by tRNA availability and the efficiency of codon-anticodon pairing) (Ikemura 1985). $F_{OP}$ is defined as the number of optimal codons divided by the sum of the number of optimal and nonoptimal codons. $F_{OP}$ takes values from 0.0 (where no optimal codons are used) to 1.0 (where only optimal codons are used).

**Codon adaptation index (CAI).** CAI is a measure of the relative adaptiveness of the codon usage of a gene towards the codon usage of highly expressed genes (Sharp *et al.* 1987), and computed as:

$$CAI = \exp\left(\frac{\sum_{i=1}^{20}\sum_{j=1}^{k_i} n_{ij} \ln x_{ij}}{\sum_{i=1}^{20}\sum_{j=1}^{k_i} n_{ij}}\right)$$

where $n_{ij}$ is the number of the $j$th codon for the $i$th amino acid, $k_i$ is the degree of codon degeneracy for the $i$th amino acid, and $x_{ij}$ is the relative adaptiveness ($R_4$) value of the $j$th codon for the $i$th amino acid in a reference set of highly expressed genes. Thus, CAI is defined as the geometric mean of the $R_4$ values, and ranges from 0.0 to 1.0. CAI can be used as a 'universal' measure of codon usage bias; that is, CAI values can be correlated with gene features other than gene expression level (e.g., G+C content, GC skew, and so on) using a reference set of genes which is representative of the bias (Carbone *et al.* 2003, 2005).

**Expression measure, $E(g)$.** Let $x_{ij}(g)$ indicate the relative codon frequency ($R_2$) value of the $j$th codon for the $i$th amino acid in the gene $g$. The codon usage difference of the gene $g$ relative to the gene group $G$ is calculated by the formula (Karlin *et al.* 1998a, 1998b):

$$B(g \mid G) = \sum_{i}^{20} p_i(g) \sum_{j}^{k_i} \left| x_{ij}(g) - x_{ij}(G) \right|$$

where $p_i(g)$ is the relative frequency of $i$th amino acid of the gene $g$. Denoted by $C$ is the collection of all protein genes, and by $H$ the putative highly expressed genes (those encoding ribosomal proteins, translation/transcription processing factors, and chaperone/degradation proteins). The general form of the expression measure is:

$$E(g) = \frac{B(g \mid C)}{B(g \mid H)}$$

A gene $g$ is deemed 'predicted highly expressed' (PHX) if $B(g|H)$ is lower than $B(g|C)$; i.e., $E(g)$ exceeds 1.0 (Karlin *et al.* 2000, 2001b, 2003, 2005). A gene $g$ is deemed 'putative alien' (PA) provided both $B(g|H)$ and $B(g|C)$ exceed the median value for all genes (Mrazek *et al.* 1999; Karlin *et al.* 2001a; Mrazek *et al.* 2001).

## EXAMPLE WORKFLOW

G-language GAE is utilized in many fields of bioinformatics and computational biology, including genomics, software development for bioinformatics, systems biology, noncoding RNAs, and *cis*-acting sequence elements (see http://www.g-language.org/wiki/publications for a list of scientific publications citing G-language GAE). Nonetheless, as described thus far, G-language GAE is especially comprehensive for the analysis of sequence conservation by information theory, genomic compositional asymmetry, and synonymous codon usage bias, in comparison with similar software packages in each of these areas. Although programming is inevitable in bioinformatics, initial screening process for these areas may be achieved only with simple combination of implemented programs within the interactive shell environment. G-language GAE supports the creation of workflows using implemented methods with minimal scripting, and several published works provide such workflow files (GCF format) as supplementary materials so that the procedure can be reused and configured (Sato *et al.* 2003; Arakawa *et al.* 2005b; Yachie *et al.* 2006).

Here we describe one example workflow that we have actually used in the screening process of a recent work that analyzed the correlation of gene positioning relative to the replication origin and the gene features (GC3, gene length, predicted gene expression level, general codon usage bias, essentiality, and functional classification) within circular bacterial chromosomes (Arakawa *et al.* 2007b). Workflow diagram for the initial screening to see if these gene features are correlated with their relative positions from replication origin is depicted in **Fig. 6**. Firstly, a GenBank format file is automatically downloaded, parsed, and loaded upon calling "load" function with appropriate RefSeq accession number of *E. coli*. Thus obtained genome data object is then annotated for the gene features of interest: gene essentiality using Profiling of *E. coli* Chromosome database (Hashimoto *et al.* 2005), GC3 of each gene, functional classification using the NCBI COG database through "set_gpac" program (Tatusov *et al.* 2001), CAI calculated using ribosomal proteins as reference (predicted gene expression level), and CAI calculated using all proteins as reference (general codon usage bias). Genes are then ordered by their relative distances from replication origin, after the prediction of replication origin and terminus using GC skew-based methods and database searches. Finally, gene features and the relative positions from the origin are statistically compared and graphed to ease the interpretation of results. G-language GAE is equipped with many basic statistics tools, so the distribution of values are first tested for normality using Kolomogorov-Smirnov Lilliefors test, and then based on this result, the degree of correlation between two variables is quantified using Spearman's rank correlation coefficient. Although further analysis in detail requires programming, all tasks in this workflow can be achieved with implemented methods only, within the interactive shell.

## CONCLUSIONS AND OUTLOOK

G-language GAE is unique among many bioinformatics workbenches, since it provides numerous genome analysis tools and algorithms in the form of programming interface, examples of those especially related to bacterial genomes analysis are described in this review. Programming is the central means in computational biology research, and availability of useful algorithms as APIs allows maximum flexibility and freedom for computational molecular biologist in combination with basic interface libraries provided by Bio* projects. As exemplified by the methods for the analysis of binding sites, codons, and nucleotide composi-

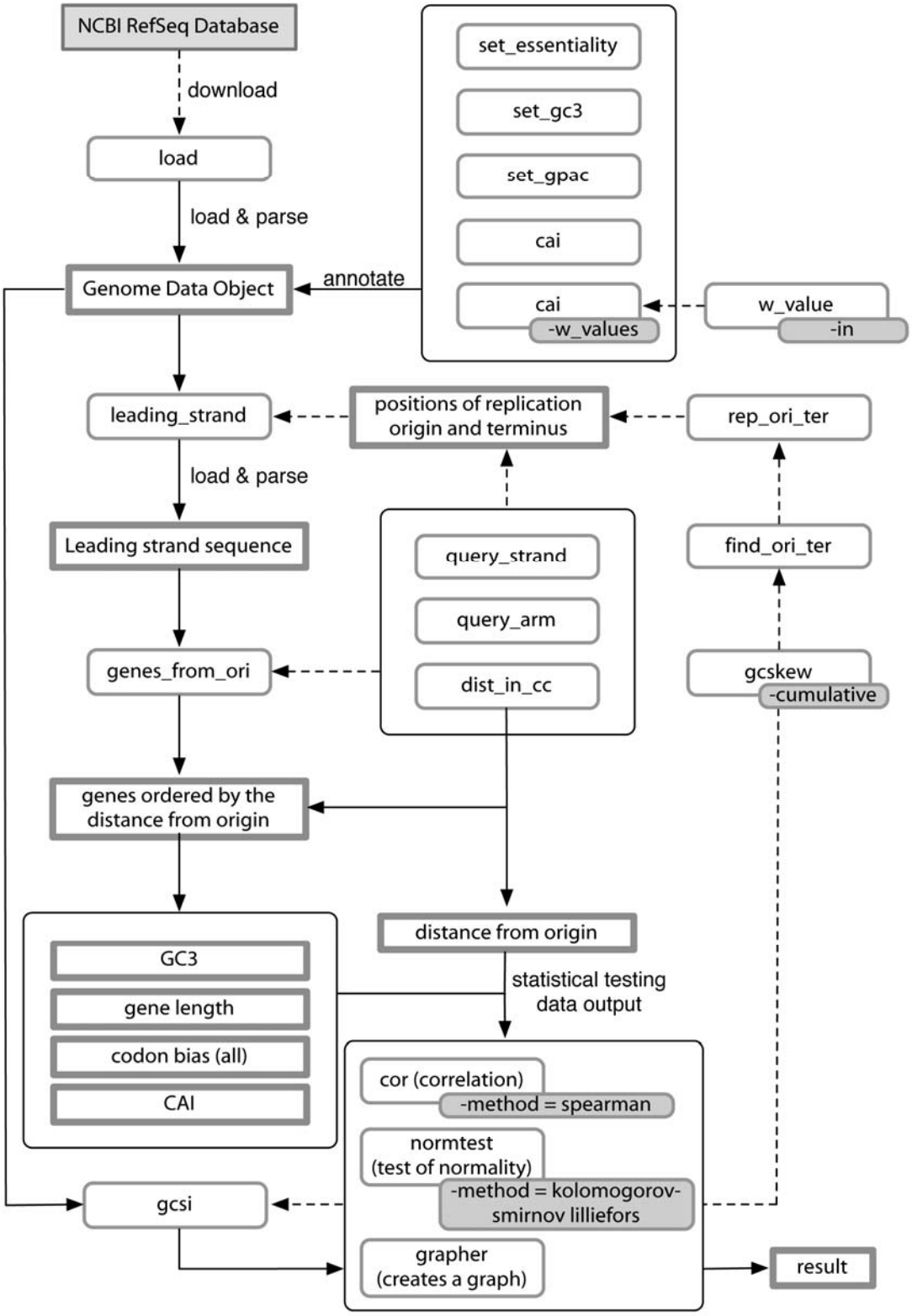


**Fig. 6 Example workflow of our recent analysis (Arakawa *et al.* 2007b).** Round-corner rectangles represent the methods implemented in G-language GAE (with grayed round-corner rectangles representing significant optional parameter), bold corner rectangles represent intermediate data, and grayed regular rectangle represents external data. Parent programs automatically use programs connected by dotted lines internally. Although programming is required for further statistical analyses, initial screening can be conducted with implemented features.

tions, G-language GAE is implemented with several algorithms for a given biological problem, therefore the users can choose the most suitable method for their needs and objectives. Most analysis programs in G-language GAE output graphical results in the form of graphs and diagrams in order to ease the interpretation by researchers, and they are also equipped with optional parameters with which analyses can be fine-tuned and configured.

We would like to stress the fact that any daily research work including computational genome analysis is mostly comprised of the trial-and-error processes, where researchers explore numerous datasets, tools and algorithms, and their parameters in combination, in order to best solve their biological problems. Therefore, to make the research process more efficient, a workbench for bioinformatics should support the heuristic nature of research routines. G-language shell interface with persistent memory, help command, logging as Perl script, tab-completion of file and program names, and interactivity coupled with the visual output and optional parameters is entirely designed for this

purpose. Workflows for bioinformatics, the result of such trial-and-errors, are essential to allow reuse and sharing of methods as is commonly done in "wet" biology with experimental protocols, but the major bottleneck in current computational biology, in our opinions, is in the heuristic *process* in the construction of workflows.

In addition to enriching more methods and documentations within the software system, we here propose three areas where bioinformatics software environment can possibly improve to make the heuristic processes more efficient: web-service, scientific visualization, and user interface. Rapid accumulation of sequence data in public databases exceeds the rate of the Moore's law of transistors (Benson *et al.* 2007), and this is expected to further accelerate with the introduction of next-generation sequencers (Blow 2007). Considering the amount of data required to transfer over internet in order to mirror locally besides the regular updates on annotations, software installation and compatibility, and wealth of computational resource typically equipped in large database providers, it is often useful to take advantage of web-services in heuristic screening (Fox *et al.* 2007). Standardization of input/output data types and method classification, redundancy in service provision, distribution of computation utilizing grid environment, and client software to support these frameworks would be necessary to make full use of the hundreds of services in concert (Stein 2002). Scientific visualization is gaining much attention in light of the complex nature of omics data and it has been successful to display results of biological research (Ball 2002; Arakawa *et al.* 2005a; Kono *et al.* 2006), but visualization that aids the heuristic processes of scientific research is less explored. Visualization that does not necessarily show the final results but that allows researchers to identify certain meanings and patterns within the huge masses of information, possibly through frequent interactions by researchers according to their heuristics, would contribute to both of computational and experimental biology. In terms of software engineering, these new frontiers should be coupled with better user interfaces that give prompt response (which may require greater computational efficiency) upon user interaction to accelerate the heuristics of researchers. As a workbench for bioinformatics, these challenges are the current goals for G-language Project.

## ACKNOWLEDGEMENTS

This research is supported in part by the Grant-in-Aid for Young Scientists (Start-up, B), No. 19810021, 2007, and No. 20710158, 2008, from the Japan Society for the Promotion of Science (JSPS), and by the funds from Yamagata Prefectural Government and Tsuruoka City.

## REFERENCES

**Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ** (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **25**, 3389-3402

**Arakawa K, Kono N, Yamada Y, Mori H, Tomita M** (2005a) KEGG-based pathway visualization tool for complex omics data. *In Silico Biology* **5**, 419-423

**Arakawa K, Mori K, Ikeda K, Matsuzaki T, Kobayashi Y, Tomita M** (2003) G-language Genome Analysis Environment: A workbench for nucleotide sequence data mining. *Bioinformatics* **19**, 305-306

**Arakawa K, Saito R, Tomita M** (2007a) Noise-reduction filtering for accurate detection of replication termini in bacterial genomes. *FEBS Letters* **581**, 253-258

**Arakawa K, Suzuki H, Fujishima K, Fujimoto K, Ueda S, Matsui M, Tomita M** (2005b) A comprehensive software suite for the analysis of cDNAs. *Genomics Proteomics Bioinformatics* **3**, 179-188

**Arakawa K, Tomita M** (2006a) G-language System as a platform for large-scale analysis of high-throughput omics data. *Journal of Pesticide Science* **31**, 282-288

**Arakawa K, Tomita M** (2006b) Large-scale modeling for systems biology. *BIOforum Europe* **10**, 54-55

**Arakawa K, Tomita M** (2007b) Selection effects on the positioning of genes and gene structures from the interplay of replication and transcription in bacterial genomes. *Evolutionary Bioinformatics* **3**, 279-286

**Arakawa K, Tomita M** (2007c) The GC skew index: a measure of genomic compositional asymmetry and the degree of replicational selection. *Evolutionary Bioinformatics* **3**, 145-154

**Ardell DH, Kirsebom LA** (2005) The genomic pattern of tDNA operon expression in *E. coli*. *PLoS Computational Biology* **1**, e12

**Ball P** (2002) Data visualization: Picture this. *Nature* **418**, 11-13

**Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J** (2008) Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics* **41**, 706-716

**Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL** (2007) GenBank. *Nucleic Acids Research* **35**, D21-D25

**Bigot S, Sivanathan V, Possoz C, Barre FX, Cornet F** (2007) FtsK, a literate chromosome segregation machine. *Molecular Microbiology* **64**, 1434-1441

**Blow N** (2007) Genomics: the personal side of genomics. *Nature* **449**, 627-630

**Brewer BJ** (1988) When polymerases collide: replication and the transcriptional organization of the *E. coli* chromosome. *Cell* **53**, 679-686

**Butler D** (2001) Data, data, everywhere. *Nature* **414**, 840-841

**Carbone A, Kepes F, Zinovyev A** (2005) Codon bias signatures, organization of microorganisms in codon space, and lifestyle. *Molecular Biology and Evolution* **22**, 547-561

**Carbone A, Zinovyev A, Kepes F** (2003) Codon adaptation index as a measure of dominating codon bias. *Bioinformatics* **19**, 2005-2015

**Cartharius K, Frech K, Grote K, Klocke B, Haltmeier M, Klingenhoff A, Frisch M, Bayerlein M, Werner T** (2005) MatInspector and beyond: promoter analysis based on transcription factor binding sites. *Bioinformatics* **21**, 2933-2942

**Carver T, Bleasby A** (2003) The design of Jemboss: A graphical user interface to EMBOSS. *Bioinformatics* **19**, 1837-1843

**Chen SL, Lee W, Hottes AK, Shapiro L, McAdams HH** (2004a) Codon usage between genomes is constrained by genome-wide mutational processes. *Proceedings of the National Academy of Science USA* **101**, 3480-3485

**Chen YT, Chang HY, Lai YC, Pan CC, Tsai SF, Peng HL** (2004b) Sequencing and analysis of the large virulence plasmid pLVPK of *Klebsiella pneumoniae* CG43. *Gene* **337**, 189-198

**Chicurel M** (2002) Bioinformatics: Bringing it all together. *Nature* **419**, 751, 753, 755 passim

**Coenye T, Vandamme P** (2003) Extracting phylogenetic information from whole-genome sequencing projects: The lactic acid bacteria as a test case. *Microbiology* **149**, 3507-3517

**Coenye T, Vandamme P** (2004) Use of the genomic signature in bacterial classification and identification. *Systematic and Applied Microbiology* **27**, 175-185

**Collins FS, Morgan M, Patrinos A** (2003) The Human Genome Project: Lessons from large-scale biology. *Science* **300**, 286-290

**Comeron JM, Aguade M** (1998) An evaluation of measures of synonymous codon usage bias. *Journal of Molecular Evolution* **47**, 268-274

**Couturier E, Rocha EP** (2006) Replication-associated gene dosage effects shape the genomes of fast-growing bacteria but only for transcription and translation genes. *Molecular Microbiology* **59**, 1506-1518

**Crooks GE, Hon G, Chandonia JM, Brenner SE** (2004) WebLogo: A sequence logo generator. *Genome Research* **14**, 1188-1190

**Daubin V, Perriere G** (2003) G+C3 structuring along the genome: A common feature in prokaryotes. *Molecular Biology and Evolution* **20**, 471-483

**Dufraigne C, Fertil B, Lespinats S, Giron A, Deschavanne P** (2005) Detection and characterization of horizontal transfers in prokaryotes using genomic signature. *Nucleic Acids Research* **33**, e6

**Eddy SR** (1998) Profile hidden Markov models. *Bioinformatics* **14**, 755-763

**Ermolaeva MD** (2001) Synonymous codon usage in bacteria. *Current Issues in Molecular Biology* **3**, 91-97

**Fox JA, McMillan S, Ouellette BF** (2007) Conducting research on the web: 2007 update for the bioinformatics links directory. *Nucleic Acids Research* **35**, W3-5

**Frank AC, Lobry JR** (1999) Asymmetric substitution patterns: A review of possible underlying mutational or selective mechanisms. *Gene* **238**, 65-77

**Frank AC, Lobry JR** (2000) Oriloc: Prediction of replication boundaries in unannotated bacterial chromosomes. *Bioinformatics* **16**, 560-561

**Freeman JM, Plasterer TN, Smith TF, Mohr SC** (1998) Patterns of genome organization in bacteria. *Science* **279**, 1827a

**Freire-Picos MA, Gonzalez-Siso MI, Rodriguez-Belmonte E, Rodriguez-Torres AM, Ramil E, Cerdan ME** (1994) Codon usage in *Kluyveromyces lactis* and in yeast cytochrome *c*-encoding genes. *Gene* **139**, 43-49

**Garcia-Vallve S, Romeu A, Palau J** (2000) Horizontal gene transfer in bacterial and archaeal complete genomes. *Genome Research* **10**, 1719-1725

**Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JY, Zhang J** (2004) Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology* **5**, R80

**Gouy M, Gautier C** (1982) Codon usage in bacteria: Correlation with gene expressivity. *Nucleic Acids Research* **10**, 7055-7074

**Grantham R, Gautier C, Gouy M, Mercier R, Pave A** (1980) Codon catalog usage and the genome hypothesis. *Nucleic Acids Research* **8**, r49-r62

**Grigoriev A** (1998) Analyzing genomes with cumulative skew diagrams. *Nucleic Acids Research* **26**, 2286-2290

**Grocock RJ, Sharp PM** (2002) Synonymous codon usage in *Pseudomonas aeruginosa* PA01. *Gene* **289**, 131-139

**Grosjean H, Fiers W** (1982) Preferential codon usage in prokaryotic genes: The optimal codon-anticodon interaction energy and the selective codon usage in efficiently expressed genes. *Gene* **18**, 199-209

**Haines AS, Akhtar P, Stephens ER, Jones K, Thomas CM, Perkins CD, Williams JR, Day MJ, Fry JC** (2006) Plasmids from freshwater environments capable of IncQ retrotransfer are diverse and include pQKH54, a new IncP-1 subgroup archetype. *Microbiology* **152**, 2689-2701

**Hashimoto M, Ichimura T, Mizoguchi H, Tanaka K, Fujimitsu K, Keyamura K, Ote T, Yamakawa T, Yamazaki Y, Mori H, Katayama T, Kato J** (2005) Cell size and nucleoid organization of engineered *Escherichia coli* cells with a reduced genome. *Molecular Microbiology* **55**, 137-149

**Hendrickson H, Lawrence JG** (2006) Selection for chromosome architecture in bacteria. *Journal of Molecular Evolution* **62**, 615-629

**Henry I, Sharp PM** (2007) Predicting gene expression level from codon usage bias. *Molecular Biology and Evolution* **24**, 10-12

**Hertzberg L, Zuk O, Getz G, Domany E** (2005) Finding motifs in promoter regions. *Journal of Computational Biology* **12**, 314-330

**Hill TM** (1992) Arrest of bacterial DNA replication. *Annual Review of Microbiology* **46**, 603-633

**Hood L, Galas D** (2003) The digital code of DNA. *Nature* **421**, 444-448

**Ideker T, Valencia A** (2006) Bioinformatics in the human interactome project. *Bioinformatics* **22**, 2973-2974

**Ikemura T** (1985) Codon usage and tRNA content in unicellular and multicellular organisms. *Molecular Biology and Evolution* **2**, 13-34

**Kaguni JM** (1997) *Escherichia coli* DnaA protein: The replication initiator. *Molecules and Cells* **7**, 145-157

**Kanaya S, Kudo Y, Nakamura Y, Ikemura T** (1996) Detection of genes in *Escherichia coli* sequences determined by genome projects and prediction of protein production levels, based on multivariate diversity in codon usage. *Computer Applications in the Biosciences* **12**, 213-225

**Karlin S** (2001) Detecting anomalous gene clusters and pathogenicity islands in diverse bacterial genomes. *Trends in Microbiology* **9**, 335-343

**Karlin S, Barnett MJ, Campbell AM, Fisher RF, Mrazek J** (2003) Predicting gene expression levels from codon biases in alpha-proteobacterial genomes. *Proceedings of the National Academy of Sciences USA* **100**, 7313-7318

**Karlin S, Burge C** (1995) Dinucleotide relative abundance extremes: a genomic signature. *Trends in Genetics* **11**, 283-290

**Karlin S, Campbell AM, Mrazek J** (1998a) Comparative DNA analysis across diverse genomes. *Annual Review of Genetics* **32**, 185-225

**Karlin S, Mrazek J** (2000) Predicted highly expressed genes of diverse prokaryotic genomes. *Journal of Bacteriology* **182**, 5238-5250

**Karlin S, Mrazek J** (2001a) Predicted highly expressed and putative alien genes of *Deinococcus radiodurans* and implications for resistance to ionizing radiation damage. *Proceedings of the National Academy of Sciences USA* **98**, 5240-5245

**Karlin S, Mrazek J, Campbell A, Kaiser D** (2001b) Characterizations of highly expressed genes of four fast-growing bacteria. *Journal of Bacteriology* **183**, 5025-5040

**Karlin S, Mrazek J, Campbell AM** (1998b) Codon usages in different gene classes of the *Escherichia coli* genome. *Molecular Microbiology* **29**, 1341-1355

**Karlin S, Mrazek J, Ma J, Brocchieri L** (2005) Predicted highly expressed genes in archaeal genomes. *Proceedings of the National Academy of Sciences USA* **102**, 7303-7308

**Kell DB, Oliver SG** (2004) Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era. *Bioessays* **26**, 99-105

**Kitano H** (2002a) Computational systems biology. *Nature* **420**, 206-210

**Kitano H** (2002b) Systems biology: A brief overview. *Science* **295**, 1662-1664

**Kono N, Arakawa K, Tomita M** (2006) MEGU: Pathway mapping web-service based on KEGG and SVG. *In Silico Biology* **6**, 621-625

**Kozak M** (1987) An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs. *Nucleic Acids Research* **15**, 8125-8148

**Lafay B, Lloyd AT, McLean MJ, Devine KM, Sharp PM, Wolfe KH** (1999) Proteome composition and codon usage in spirochaetes: Species-specific and DNA strand-specific mutational biases. *Nucleic Acids Research* **27**, 1642-1649

**Lawrence JG, Ochman H** (1997) Amelioration of bacterial genomes: Rates of change and exchange. *Journal of Molecular Evolution* **44**, 383-397

**Levy O, Ptacin JL, Pease PJ, Gore J, Eisen MB, Bustamante C, Cozzarelli NR** (2005) Identification of oligonucleotide sequences that direct the movement of the *Escherichia coli* FtsK translocase. *Proceedings of the National Academy of Sciences USA* **102**, 17618-17623

**Liolios K, Tavernarakis N, Hugenholtz P, Kyrpides NC** (2006) The Genomes On Line Database (GOLD) v.2: A monitor of genome projects worldwide. *Nucleic Acids Research* **34**, D332-334

**Liu B, Alberts BM** (1995) Head-on collision between a DNA replication apparatus and RNA polymerase transcription complex. *Science* **267**, 1131-1137

**Lobry JR** (1996) Asymmetric substitution patterns in the two DNA strands of bacteria. *Molecular Biology and Evolution* **13**, 660-665

**Lobry JR, Louarn JM** (2003) Polarisation of prokaryotic chromosomes. *Current Opinion in Microbiology* **6**, 101-108

**Lobry JR, Necsulea A** (2006) Synonymous codon usage and its potential link with optimal growth temperature in prokaryotes. *Gene* **385**, 128-136

**Lobry JR, Sueoka N** (2002) Asymmetric directional mutation pressures in bacteria. *Genome Biology* **3**, RESEARCH0058

**Lynn DJ, Singer GA, Hickey DA** (2002) Synonymous codon usage is subject to selection in thermophilic bacteria. *Nucleic Acids Research* **30**, 4272-4277

**Mangalam H** (2002) The Bio* toolkits – a brief overview. *Briefings in Bioinformatics* **3**, 296-302

**McInerney JO** (1998) Replicational and transcriptional selection on codon usage in *Borrelia burgdorferi*. *Proceedings of the National Academy of Sciences USA* **95**, 10698-10703

**McLean MJ, Wolfe KH, Devine KM** (1998) Base composition skews, replication orientation, and gene orientation in 12 prokaryote genomes. *Journal of Molecular Evolution* **47**, 691-696

**Medigue C, Rouxel T, Vigier P, Henaut A, Danchin A** (1991) Evidence for horizontal gene transfer in *Escherichia coli* speciation. *Journal of Molecular Biology* **222**, 851-856

**Morton BR** (1993) Chloroplast DNA codon use: evidence for selection at the psb A locus based on tRNA availability. *Journal of Molecular Evolution* **37**, 273-280

**Moszer I, Glaser P, Danchin A** (1991) Multiple IS insertion sequences near the replication terminus in *Escherichia coli* K-12. *Biochimie* **73**, 1361-1374

**Moszer I, Rocha EP, Danchin A** (1999) Codon usage and lateral gene transfer in *Bacillus subtilis*. *Current Opinion in Microbiology* **2**, 524-528

**Mrazek J, Bhaya D, Grossman AR, Karlin S** (2001) Highly expressed and alien genes of the *Synechocystis* genome. *Nucleic Acids Research* **29**, 1590-1601

**Mrazek J, Karlin S** (1999) Detecting alien genes in bacterial genomes. *Annals of the New York Academy of Science* **870**, 314-329

**Muto A, Osawa S** (1987) The guanine and cytosine content of genomic DNA and bacterial evolution. *Proceedings of the National Academy of Sciences USA* **84**, 166-169

**Naya H, Romero H, Zavala A, Alvarez B, Musto H** (2002) Aerobiosis increases the genomic guanine plus cytosine content (GC%) in prokaryotes. *Journal of Molecular Evolution* **55**, 260-264

**Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock MR, Wipat A, Li P** (2004) Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* **20**, 3045-3054

**Omont N, Kepes F** (2004) Transcription/replication collisions cause bacterial transcription units to be longer on the leading strand of replication. *Bioinformatics* **20**, 2719-2725

**Pease PJ, Levy O, Cost GJ, Gore J, Ptacin JL, Sherratt D, Bustamante C, Cozzarelli NR** (2005) Sequence-directed DNA translocation by purified FtsK. *Science* **307**, 586-590

**Perals K, Capiaux H, Vincourt JB, Louarn JM, Sherratt DJ, Cornet F** (2001) Interplay between recombination, cell division and chromosome structure during chromosome dimer resolution in *Escherichia coli*. *Molecular Microbiology* **39**, 904-913

**Perals K, Cornet F, Merlet Y, Delon I, Louarn JM** (2000) Functional polarization of the *Escherichia coli* chromosome terminus: the dif site acts in chromosome dimer resolution only when located between long stretches of opposite polarity. *Molecular Microbiology* **36**, 33-43

**Perriere G, Thioulouse J** (2002) Use and misuse of correspondence analysis in codon usage studies. *Nucleic Acids Research* **30**, 4548-4555

**Price MN, Alm EJ, Arkin AP** (2005) Interruptions in gene expression drive highly expressed operons to the leading strand of DNA replication. *Nucleic Acids Research* **33**, 3224-3234

**Rice P, Longden I, Bleasby A** (2000) EMBOSS: The European Molecular Biology Open Software Suite. *Trends in Genetics* **16**, 276-277

**Rocha EP** (2004a) Codon usage bias from tRNA's point of view: Redundancy, specialization, and efficient decoding for translation optimization. *Genome Research* **14**, 2279-2286

**Rocha EP** (2004b) Order and disorder in bacterial genomes. *Current Opinion in Microbiology* **7**, 519-527

**Rocha EP** (2004c) The replication-related organization of bacterial genomes. *Microbiology* **150**, 1609-1627

**Rocha EP, Danchin A** (2002) Base composition bias might result from competition for metabolic resources. *Trends in Genetics* **18**, 291-294

**Rocha EP, Danchin A** (2003) Essentiality, not expressiveness, drives gene-strand bias in bacteria. *Nature Genetics* **34**, 377-378

**Sato M, Umeki H, Saito R, Kanai A, Tomita M** (2003) Computational analysis of stop codon readthrough in *D. melanogaster*. *Bioinformatics* **19**, 1371-1380

**Schneider G, Fechner U** (2004) Advances in the prediction of protein targeting signals. *Proteomics* **4**, 1571-1580

**Schneider TD** (1997) Information content of individual genetic sequences. *Journal of Theoretical Biology* **189**, 427-441

**Schneider TD** (1999) Measuring molecular information. *Journal of Theoretical Biology* **201**, 87-92

**Schneider TD** (2002) Consensus sequence Zen. *Applied Bioinformatics* **1**, 111-119

**Schneider TD, Stephens RM** (1990) Sequence logos: A new way to display consensus sequences. *Nucleic Acids Research* **18**, 6097-6100

**Schuster-Bockler B, Schultz J, Rahmann S** (2004) HMM Logos for visualization of protein families. *BMC Bioinformatics* **5**, 7

**Shannon CE** (1948) A mathematical theory of communication. *Bell System Technical Journal* **27**, 379-423, 623-656

**Sharp PM, Bailes E, Grocock RJ, Peden JF, Sockett RE** (2005) Variation in the strength of selected codon usage bias among bacteria. *Nucleic Acids Research* **33**, 1141-1153

**Sharp PM, Cowe E, Higgins DG, Shields DC, Wolfe KH, Wright F** (1988) Codon usage patterns in *Escherichia coli*, *Bacillus subtilis*, *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe*, *Drosophila melanogaster* and *Homo sapiens*; a review of the considerable within-species diversity. *Nucleic Acids Research* **16**, 8207-8211

**Sharp PM, Li WH** (1987) The codon adaptation index - a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research* **15**, 1281-1295

**Sharp PM, Tuohy TM, Mosurski KR** (1986) Codon usage in yeast: Cluster analysis clearly differentiates highly and lowly expressed genes. *Nucleic Acids Research* **14**, 5125-5143

**Shields DC, Sharp PM** (1987) Synonymous codon usage in *Bacillus subtilis* reflects both translational selection and mutational biases. *Nucleic Acids Research* **15**, 8023-8040

**Shine J, Dalgarno L** (1974) The 3'-terminal sequence of *Escherichia coli* 16S ribosomal RNA: complementarity to nonsense triplets and ribosome binding sites. *Proceedings of the National Academy of Sciences USA* **71**, 1342-1346

**Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, Fuellen G, Gilbert JG, Korf I, Lapp H, Lehvaslaiho H, Matsalla C, Mungall CJ, Osborne BI, Pocock MR, Schattner P, Senger M, Stein LD, Stupka E, Wilkinson MD, Birney E** (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Research* **12**, 1611-1618

**Stein L** (1996) How Perl Saved the Human Genome Project. *The Perl Journal* **1**, 5-9

**Stein L** (2002) Creating a bioinformatics nation. *Nature* **417**, 119-120

**Stormo GD, Schneider TD, Gold L, Ehrenfeucht A** (1982) Use of the 'Perceptron' algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic Acids Research* **10**, 2997-3011

**Sueoka N** (1962) On the genetic basis of variation and heterogeneity of DNA base composition. *Proceedings of the National Academy of Sciences USA* **48**, 582-592

**Suzuki H, Saito R, Tomita M** (2004) The 'weighted sum of relative entropy': A new index for synonymous codon usage bias. *Gene* **335**, 19-23

**Suzuki H, Saito R, Tomita M** (2005) A problem in multivariate analysis of codon usage data and a possible solution. *FEBS Letters* **579**, 6499-6504

**Swertz MA, Jansen RC** (2007) Beyond standardization: Dynamic software infrastructures for systems biology. *Nature Review Genetics* **8**, 235-243

**Tatusov RL, Natale DA, Garkavtsev IV, Tatusova TA, Shankavaram UT, Rao BS, Kiryutin B, Galperin MY, Fedorova ND, Koonin EV** (2001) The COG database: New developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Research* **29**, 22-28

**Thompson JD, Higgins DG, Gibson TJ** (1994) CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* **22**, 4673-4680

**van Passel MW, Bart A, Thygesen HH, Luyf AC, van Kampen AH, van der Ende A** (2005) An acquisition account of genomic islands based on genome signature comparisons. *BMC Genomics* **6**, 163

**van Passel MW, Kuramae EE, Luyf AC, Bart A, Boekhout T** (2006) The reach of the genome signature in prokaryotes. *BMC Evolutionary Biology* **6**, 84

**von Samson-Himmelstjerna G, Harder A, Failing K, Pape M, Schnieder T** (2003) Analysis of codon usage in beta-tubulin sequences of helminths. *Parasitology Research* **90**, 294-300

**Walker DR, Koonin EV** (1997) SEALS: A system for easy analysis of lots of sequences. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology* **5**, 333-339

**Wilkinson M, Schoof H, Ernst R, Haase D** (2005) BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case. *Plant Physiology* **138**, 5-17

**Wright F** (1990) The 'effective number of codons' used in a gene. *Gene* **87**, 23-29

**Yachie N, Numata K, Saito R, Kanai A, Tomita M** (2006) Prediction of non-coding and antisense RNA genes in *Escherichia coli* with Gapped Markov Model. *Gene* **372**, 171-181

**Zeeberg B** (2002) Shannon information theoretic computation of synonymous codon usage biases in coding regions of human and mouse genomes. *Genome Research* **12**, 944-955