PLATCOM: Current Status and Plan for the Next Stages

Kwangmin Choi [a] Jeong-Hyeon Choi [a] Sun Kim [a,b,*,1,2,3]

[a] *School of Informatics, Indiana University, IN 47408, USA*
[b] *Center for Genomics and Bioinformatics, Indiana University, IN 47405, USA*

Abstract

We have been developing a system for comparing multiple genomes, PLATCOM, where users can choose genomes of their choice freely and perform analysis of the selected genomes with a suite of computational tools. PLATCOM is built on internal databases such as GenBank, COG, KEGG, and Pairwise Comparison Database (PCDB) that contains all pairwise comparisons (97,034 entries) of protein sequence files (.faa) and whole genome sequence files (.fna) of 312 replicons. PCDB is designed to incorporate new genomes automatically, so that PLATCOM can evolve as new genomes become available. PLATCOM is available at `http://platcom.informatics.indiana.edu`.

The design goal of PLATCOM is to provide a flexible environment for comparison of genomes from the "sequence analysis perspective." Comparison of multiple genomes is a challenging task since combining multiple tools for sequence analysis requires a significant amount of programming work and knowledge on each tool. To alleviate such problem, we borrowed techniques from existing systems, and we have also developed and incorporated high performance sequence data mining tools such as sequence clustering and neighborhood prediction. High performance data mining tools have been useful in integrating separate system modules by gluing them together on the biological sequence level.

PLATCOM is designed to evolve through three development stages. Its first stage is complete: the underlying architecture and individual system modules. We share our experience in designing and implementing PLATCOM and then discuss our current design strategies that have been refined from our experience after the completion of the first implementation stage.

Key words: genome comparison, sequence analysis, data mining, integration

1 Introduction

The exponential accumulation of genomic sequence data demands systematic analysis of genetic information and requires use of various computational approaches to handle such huge sets of genomic data. Comparative genomics, with such organized data and diverse computational techniques, has become useful not only for finding common features in different genomes, but also for understanding evolutionary process and mechanism among multiple genomes.

In this paper, we will consider multiple genome comparison only from the "sequence analysis perspective." Even in this case, comparison of multiple genomes is a challenging task since combining multiple tools for sequence analysis requires a significant amount of programming work and knowledge on each tool. In particular, challenges are as follows. First of all, genome comparison involves a large amount of data and it is computationally demanding since the basic unit is a genome, e.g., the entire set of predicted proteins in the genome. Second, the choice of genomes to be compared is entirely subjective, so there are simply too many choices. For example, there are 1,313,400 ($= \binom{200}{3}$) possible selections of three genomes out of 200 completely sequenced genomes. Third, genome comparison generates a large amount of output which is hard to interpret when a raw data is presented. Thus the result should be presented with a summary, typically in a visualization format. Lastly, there are so many data sources that can be used as input to the analysis of genomes or that can be referenced by the analysis result.

Considering all these issues, it is not possible to perform multiple genome comparison by simply using sequence analysis tools in an *ad hoc* fashion. Thus, there has been significant research on building such systems. In the next section, we briefly survey systems that can be used for genome comparison from the sequence analysis perspective: SEALS, The SEED, DAS, BioWorks, and MBGD. Section 3 describes PLATCOM, a genome comparison system of our own, and then Section 4 shares our experience in developing PLATCOM. We propose a design paradigm for genome comparison systems in Section 5. Then conclusion follows.

* Corresponding authors.
Email addresses: kwchoi@indiana.edu (Kwangmin Choi),
jeochoi@indiana.edu (Jeong-Hyeon Choi),
sunkim@bio.informatics.indiana.edu (Sun Kim).
[1] Supported by NSF DBI-0237901.
[2] Supported by NSF 0116050.
[3] Supported by INGEN (Indiana Genomics Initiatives)

2 A survey of systems for multiple genome comparison

System for Easy Analysis of Lots of Sequences (SEALS) [1] is a comparative genome analysis system designed to facilitate sequence analysis projects that handle huge amounts of genome data. SEALS provides modular components which can be combined, modified, and integrated with other modules in order to quickly design and execute in silico experiments for sequence analysis projects at the scale of whole genomes. These modules can be used in a way similar to UNIX-style command-line environment. Wrappers are also provided to combine commonly used analysis programs.

The SEED [2] aims to provide a suite of programs which enable distributed users to annotate new genomes rapidly and cooperatively. By using this system, users can create, collect, and maintain sets of gene annotations organized by groups of related biological and biochemical functions (called, "subsystems") among many genomes. The subsystem is defined as a set of biological functions that together implement a specific process. By annotating one subsystem at a time, the SEED supports the annotation of a single subsystem over multiple genomes simultaneously. So users may examine the relationship between a given gene and a group of other genes by using contextual clues relevant to the determination of functions.

The distributed annotation system (DAS) [3] is a genome annotation system where multiple third-party groups can annotate the genome sequence with a combination of computational and experimental methods using diverse analytic tools and data models. To hanlde information fragmentation and inconsistency, it employs a client-server approach so that a client accesses genome annotation information from the multiple distant reference and annotation servers, collates that information, and displays it to the user in a single view. DAS is designed to allow sequence annotation to be decentralized and integrated on an as-needed basis by client-side program. When a server is designated the reference server, it serves as third-party annotation servers and users may search against one or more annotation servers to retrieve information from a genome region of interest by using web browser-like sequence browser.

BioWorks [5] is an open source platform which was evolved from a project originally sponsored by the National Cancer Institute Center for Bioinformatics (NCICB). BioWorks is written in JAVA language and aims to provide sophisticated methods for data management, analysis and visualization. This system especially emphasize on data and algorithm integration, microarray data analysis, metabolic pathway analysis, sequence analysis, reverse engineering, transcription factor binding site detection, and motif/pattern discovery and all these features are fully implemented in the updated version.

Microbial Genome Database for Comparative Analysis (MBGD) [4] is a workbench for comparative analysis of microbial genomes, which aims at providing a classification system rather than retrieving already classified information. The core components of MBGD include (i) a gene classification algorithm into orthologous groups using pre-computed all-against-all homology search results, (ii) an intuitive user interface which helps users analyze search results, and (iii) an updating process which enables the system to provide the latest data rapidly. Users can easily create their own orthologous classification table by choosing sets of genomes. In MBGD, similarity relationships among all protein coding genes in the whole set of genomes are pre-computed and stored as a database. Users can dynamically create orthologous classification table using this pre-computed data, simply by selecting genomes and/or setting parameter values.

3 PLATCOM: A computational environment for comparative genomics

We have been developing a genome comparison system PLATCOM, which is available at `http://platcom.informatics.indiana.edu`. Its design principles are as follows:

(1) *Flexibility:* In sequence analysis, decision on which genomes to be compared or on criteria for sequence matching, i.e., cutoff thresholds, is entirely subjective. Thus the system should allow the maximum freedom on this decision.
(2) *Easy to use:* Interface to each analysis module should be simple and intuitive enough so that users can compare genomes simply by selecting genomes to be compared.
(3) *Easy to maintain and update:* The system should be designed to incorporate new genomes easily as they become available.
(4) *Reconfigurability:* Interface to system modules is defined from the sequence analysis perspective so that modules can be combined whenever possible.

These design principles may conflict with other desirable system features such as information richness and sophisticated user interface. Instead, PLATCOM aims at a flexible, extensible, scalable, and reconfigurable system with emphasis on high-performance data mining. Although PLATCOM does not store or maintain any information on sequences, information on sequences can be obtained via URL or connectivity tool to other information rich databases.

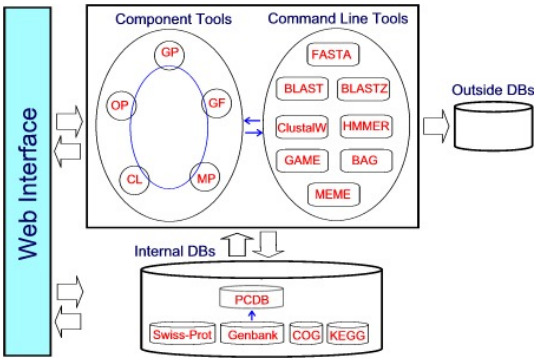


Fig. 1. Overall Architecture of PLATCOM

3.1 Overall system architecture

PLATCOM consists of four main components as shown in Figure 1:

(1) databases,
(2) sequence analysis tools,
(3) genome analysis modules, and
(4) user interfaces.

The whole system is built on internal databases, which consist of GenBank (`ftp://ftp.ncbi.nlm.nih.gov/genomes`), SwissProt (`http://www.ebi.ac.uk/swissprot`), COG (`http://www.ncbi.nlm.nih.gov/COG`), KEGG (`http://www.genome.ad.jp/kegg`), and Pairwise Comparison Database (PCDB). PCDB is designed to incorporate new genomes automatically so that PLATCOM can evolve as new genomes become available. FASTA and BLASTZ are used to compute all pairwise comparisons (97,034 entries) of protein sequence files (.faa) and whole genome sequence files (.fna) of 312 replicons. Multiple genome comparisons usually take too much time to compute, but the pre-computed PCDB makes it possible to complete genome analysis very fast even on the web. In general, PLATCOM runs several hundred times faster than a system without PCDB when several genomes are compared. In addition to sequence data, PLATCOM will include more data types such as gene expression data.

Sequence analysis tools include widely used public tools such as FASTA, BLAST, BLASTZ, HMMER, GIBBS, and MEME. We have also included our own tools such as a genome sequence alignment tool GAME [11], a sequence clustering algorithm BAG [8] and a correlated gene set mining tool [13]. We plan to include more tools.

With the databases and sequence analysis tools, genome can be compared. There are currently six modules: genome plot, conserved gene neighborhood navigation, metabolic pathways, comparative sequence clustering analysis, pu-

tative gene fusion events detection, and multiple genome alignment. These genome analysis modules can be initiated using a genome selection user interface.

3.2 Genome analysis modules

Six sequence analysis tools are embedded in the system as of March 2005. These modules are designed to be combined flexibly in a way that output from one module can be input to another module (see Section 4.3). A set of genomes selected by users is submitted with parameter settings via web interface.

- **Genome Plot:** GenomePlot is a visualization tool to generate a genome comparison diagonal plot between two selected genomes. It retrieves pairwise comparison data from pre-computed PCDB to generate 2-dimensional plot and its image map. GenomePlot provides a strong intuition to understand the overall genome structure and phylogenetic distance between two given genomes. It is also an effective way to visually identify gene clusters that are conserved between two close genomes. Further analysis on the 2D plot is allowed (see Section 4.3).

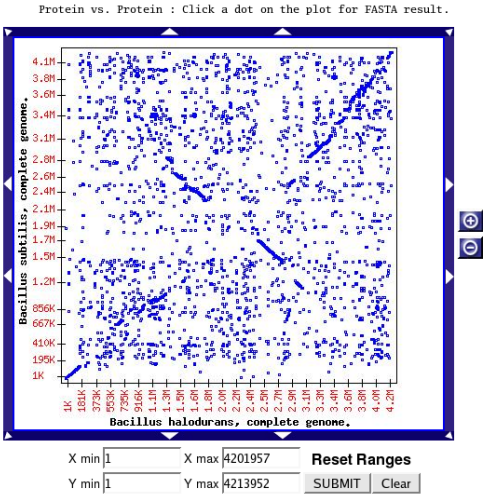


Fig. 2. GenomePlot

- **Operon Analysis:** OperonViz is a tool to navigate and visualize gene neighborhoods. Two versions of OperonViz are embedded in the system; OperonViz-COG uses COG database to identify homologs and OperonViz-BAG uses PCDB and the BAG clustering algorithm for the same purpose. If the distance is shorter than a given value (Default value is 200bp), two genes are considered to belong to the same gene clusters. OperonViz is useful to identify horizontal gene transfers, functional coupling and functional hitchhiking.

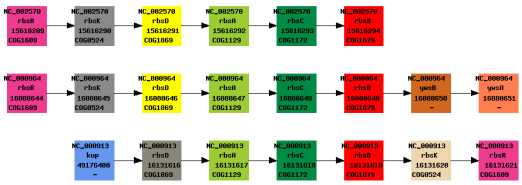


Fig. 3. OperonViz

- **Gene Fusion Event Detection:** FuzFinder uses PCDB to identify plausible gene fusion events among a set of submitted genomes. The definition of mutual best hit is as follows: (i) Each of the two reference genes must match the same open reading frame (ORF) in the target genome with a higher Z-score than a given value. (ii) When split between the two hits, the two halves of the target ORF must match back to the original two reference genes with a higher Z-score than a given value. (iii) The reference genes must not be homologous to each other. Although Z-score is a statistical score that depends on the database size, users can use the default value as genomes are fairly large. An option is provided to change the Z-score cutoff for pairwise matches.

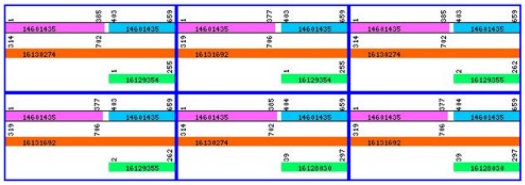
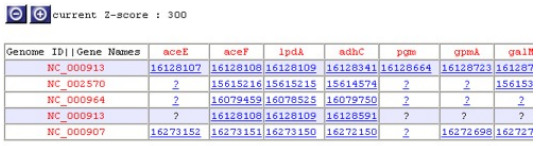


Fig. 4. FuzFinder

- **Metabolic Pathway Analysis:** MetaPath is a metabolic pathway analysis tool. It combines metabolic pathway information at KEGG and sequence information at GenBank to reconstruct metabolic pathways among the selected genomes. This tool aims to find missing genes in metabolic pathways by comparing reference genome with a set of genome selection. The result is represented as a gene-genome table, but the directionality of metabolic pathway is not considered as of now because of the lack of such information in KEGG database. MetaPath web service is limited to only prokaryotic genomes.


current Z-score : 300

Genome ID||Gene Names	aceE	aceF	lpdA	adhC	pgm	gpmA	galM
NC_000913	16128107	16128108	16128109	16128341	16128664	16128723	161287
NC_002570	?	15615216	15615215	15614574	?	?	156153
NC_000964	?	16079459	16078525	16079750	?	?	?
NC_000913	?	16128108	16128109	16128591	?	?	?
NC_000907	16273152	16273151	16273150	16272150	?	16272698	162727

Fig. 5. A gene-genome table in MetaPath genome analysis.

- **Gene Clustering Tools:** Users can upload a set of protein sequences in FASTA format using FASTA-BAG and BLASTP-BAG or select genomes from the genome list using Genome-BAG service (Genome-BAG) for clustering analysis using BAG

- **Genome Alignment Tools:** Two interfaces are provided for aligning genomic DNA sequences: a pairwise sequence alignment tool, GAME [11], and a multiple genome alignment tool [12]. GAME is a fast genome sequence alignment tool, especially effective for detecting protein coding regions. Users can input their own genomic DNA sequences and compare it against any subset of genomes out of 312 replicons. The multiple genome alignment is generated by combining precomputed pairwise genome alignment using Blastz [14]. Due to the precomputed alignment databases, users can select any subset of genomes and align them on the web.

4 Our Experience with Developing PLATCOM

In this section, we share our experience in developing PLATCOM for about two years.

4.1 Computing and updating the precomputed pairwise databases

There are several systems that precompute pairwise sequence comparisons and sotre them as databases. e.g., MBGD and The SEED. PLATCOM also precomputes and stores the pairwise comparison databases (PCDB) on protein and genomic DNA sequence levels. PCDB makes it possible to achieve one of the most important design goals, which is to allow users to select any subset of genomes to be compared freely.

An important issue is how to update PCDB. We believe that any system with PCDB has some systematic strategy for updating PCDB. However, we were not able to find any document about this updating strategy. Thus, we take this opportunity to describe our strategy. There are three systems involved in updating PCDB:

(1) the NCBI ftp site,
(2) the machine where PLATCOM is served, and
(3) a high performance machine AVIDD Linux cluster where actual pairwise comparisons are performed.

One design philosophy is to avoid any burden to synchronize information on the three machines, so we do not use the typical metadata approach. Instead, PLATCOM exactly follows the same directory structure as in the NCBI ftp site, where each genome has a separate directory. Whenever we decided to bring in a new genome N from NCBI to PLATCOM, the corresponding directory is `ftp`ed to a directory with exactly the same name as in NCBI and

the same directory hierarchy. Let us N_{dir} the directory name. Let $N_{id}.faa$ and $N_{id}.fna$ the files for proteins and nucleotide sequences respectively. Then a directory $N_{id}.faa.cmp$ is created for protein sequence comparisons and another directory $N_{id}.fna.cmp$ is created for nucleotide sequence comparisons.

Then a list of pairwise comparison jobs and a list of related files are generated by browsing the directory hierarchy. These list of jobs and related files are transfered automatically using `ssh` to the AVIDD clusters and then each job is submitted using our job management system [10]. As soon as the pairwise comparison result is available, it is automatically copied back to the corresponding directories, $N_{id}.faa$ and $N_{id}.fna$, on the PLATCOM system.

The entire procedure can be performed almost automatically and there is no need for updating any meta data for adding and deleting genomes.

4.2 Need for scalable data mining tools

Data mining tools are very important for genome comparison due to the large volume of genome data. In a sense, comparative genomics community has actively adopted sophisticated and powerful data mining techniques, because the concept of biological 'contextual information,' such as operons and metabolic pathways, corresponds to the 'context' used in the data mining community. One example is the concept of 'subsystem' in The SEED, which represents a set of sequences. Data mining techniques are also useful in combining many sequence analysis tools and databases that can be utilized for genome annotation since data mining tools encapsulate multiple sequence analysis tasks in a single step. Thus, well defined data mining concept and tools can make genome comparison much easier. For example, we have developed a system, CLASSEQ [7], where users can input uncharacterized sequences and compare them against genomes of their choice are using a sequence clustering algorithm BAG [8]. Alternatively, users can perform separate database searches with each sequence as a query against each genome, and then collect all the search results, creating clusters of sequences where any user sequences are included. Use of a clustering algorithm can make the entire task much simpler. Such simplification of the task also makes it easy to provide an interface for performing further analysis on each cluster such as common domain search and multiple sequence alignments. In addition, high level 'context' or 'abstract' makes the system much easier to understand and more robust.

It is also important that the data mining tools for genome comparison should be scalable. We have been developing such scalable tools: a sequence clustering algorithm BAG [8], an algorithm for mining correlated gene sets [13], and a multiple genome sequence alignment algorithm by clustering local matches

[12].

4.3 Multi-step sequence analysis with data visualization

Each genome analysis task generates a huge amount of result, so it is important to use visualization technique to summarize the analysis result. Thus we have developed visualization tools for genome plot, multidomain, gene-genome matching table, and genome alignment. Since our ultimate goal is to make PLATCOM a flexible system in that users can combine multiple computational tasks freely, it is also important to make visualization modules independent of particular computational tasks. We designed the interface of the visualization modules to use genomes as context so that output from different computational tasks can use the same visualization module. We will elaborate our discussion with an example in Figure 6. Users can select a pair of genomes, generating a 2D genome plot. From the genome plot, users can predict gene clusters using MCGS [13] or genome segments using FISH [9], which can be represented as a set of matching gene pairs. Each gene cluster can then be sought in other genomes, producing either a gene-genome table or gene neighborhood navigation figure on the web. The gene-genome table is a visualization interface used in MetaPATH that looks for existence of a metabolic pathway from KEGG. The gene neighborhood navigation figure is a navigation system used for OperonViz, operon visualization and navigation, using wither COG or BAG clusters. If we have well-defined interfaces, e.g., a set of genes, to the gene-genome table or the gene neighborhood navigation system, it is always possible to utilize these system modules whenever possible. We have tried to define such formal interface to each visualization systems, which led us to refine our design strategies as described in the next section.

5 Plan for the Next Stages

Most systems are not designed to perform a series of sequence analyses, which will be referred as *multi-step analysis*. The main challenge is that there are numerous ways to combine tools and databases and each multi-step analysis should be provided as a separate interface. The module-based systems, such as SEALS, DAS, The SEED, and PLATCOM, try to address this problem by providing a set of library modules so that users can combine them to provide a new multi-step analysis. However, this approach requires a substantial programming practice by someone who is already familiar with the modules. Thus the module-based approach is limited only to bioinformatics experts. The real challenge is how we can provide an environment where users (biologists or medical scientists) can perform multi-step analysis in a flexible way. Below

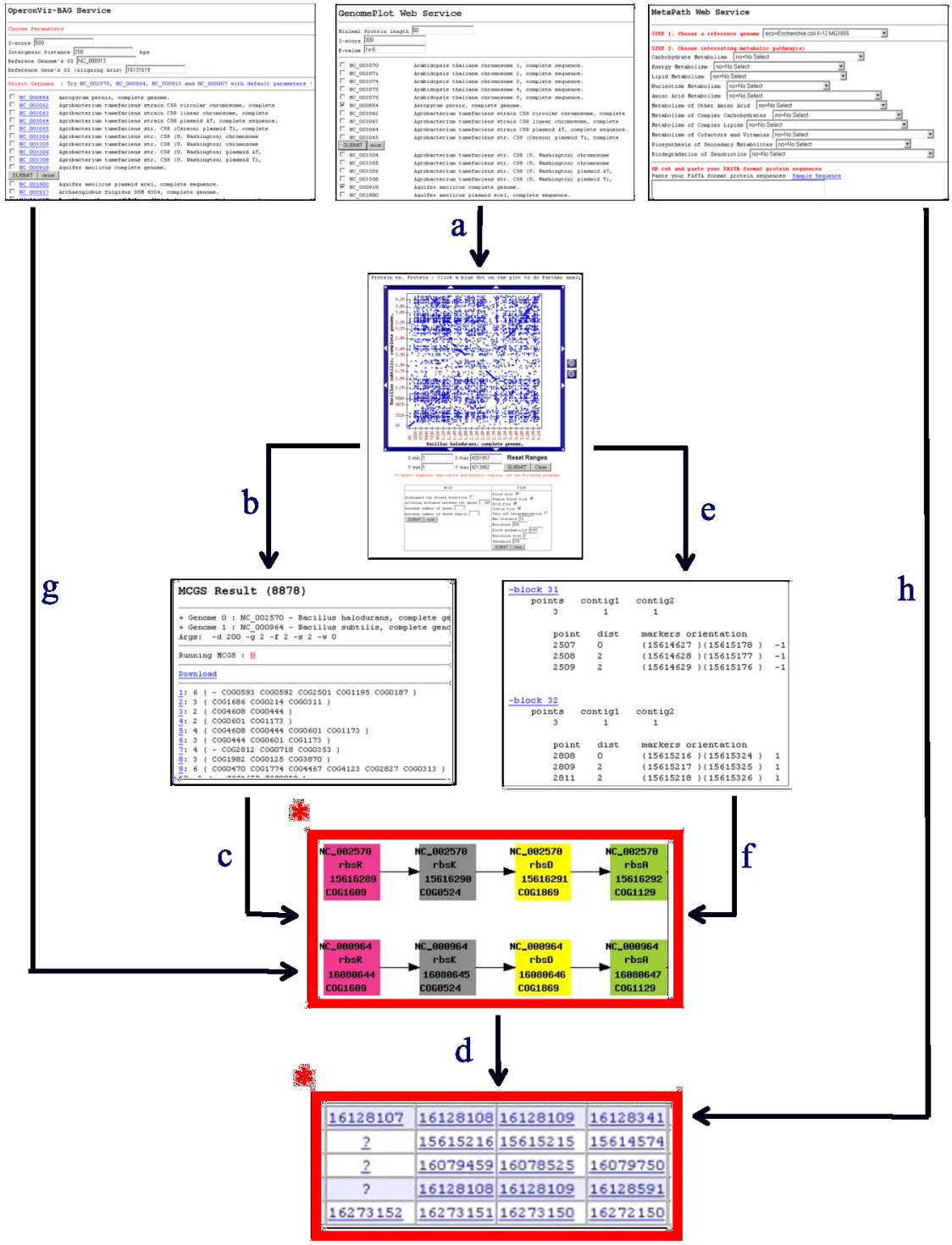


Fig. 6. An example for multi-step sequence analysis with data visualization. With consistent interfaces to the gene neighborhood navigation and the gene-genome table (screen shots with *), they can be invoked as long as the input becomes available. For example, the gene neighborhood navigation is invoked by either OperonViz user interface ($\downarrow g$), MCGS result ($c \downarrow$), or FISH result ($\downarrow f$). The gene-genome table module can be invoked by either MetaPath user interface ($h \downarrow$) or the gene neighborhood navigation module ($d \downarrow$).

we propose an approach to developing such flexible systems: defining abstract data types for genome comparison, developing high performance data mining tools, and designing and implementing genome analysis language. These concepts have been implemented in many existing systems and are not entirely new. What we are trying to achieve is to bring together useful concepts from existing systems and make them formal so that the genome comparison system can be developed with a clear conceptual design.

5.1 Data types for genome comparison

We propose "defining data types for genome comparison" as a first step toward such flexible systems. To elaborate on the genome data type, we will use the pairwise genome comparison plot function that is available in many systems. By simply selecting a pair of genomes, a two dimensional plot that shows all matching genes between the two genomes will be shown. Some systems allow users to perform further analysis on matching genes in the genome plot. For example, users can retrieve the protein sequences, search for domains in domain databases with the sequences as query, perform a pairwise alignment of the two matching sequences. In this example, users selected "genomes" and generated "genome plot" of "a pair of genomes" where each genome is a set of "gene sequences." A genome plot is a set of "pair of gene sequences" in a two dimensional plot. Now we observe that this analysis already used several important data types: genomes, a pair of genomes, gene, and a pair of genes. There are also operations on these data types. Comparison of two genomes, `genome-pair-comparison(`G_i`,` G_j`)`, is performed and a set of gene matches are generated, `pair-of-genes(`g_{i_k}`,` g_{j_l}`)`. Any two matching genes can be aligned using a pairwise sequence alignment tool, `align-two-sequence(`g_{i_k}`,` g_{j_l}`)`. In addition, each gene can be used as a query against the domain database, `search-domain(`g_{i_j}`, domain-database)`.

As in the above example, we can define abstract data types for genome comparisons. Below are a list of data types.

- sequence $\int$: A sequence can be either a protein, a genomic DNA, mRNA, or ncRNA. A subsequence of a sequence is also a sequence. Thus any subsequence of a protein, a genomic DNA, mRNA, or ncRNA is also a sequence. The entire sequence of a replicon (chromosome or plasmid) is also a sequence.
- a pair of sequences $\mathcal{P}_s$: A pair of sequences is two sequences that are matched via some sequence analysis tool.
- a set of sequences $\mathcal{S}_s$: a set of sequences is a set of sequences. The typical example would be a family of sequences.
- alignment $\mathcal{A}$: A set of aligned sequences.

- model $\mathcal{M}$: A model from a set of sequences, e.g., a profile-HMM.
- the neigborhood of sequence $\mathcal{N}$: The neigborhood of a sequence is a set of sequences within a certain physical distance in a replicon (chromosome or plasimid).
- a collection of sequences $\mathcal{C}$: A group of sequences is a set of sequences. Examples include operons that are controlled by a single promotor and a set of genes that involved in a metabolic pathway.
- replicon $\mathcal{R}$: A replicon can be either a nucleotide sequence or the entire set of gene sequences in the replicon.
- genome $\mathcal{G}$: A set of replicons. For an organism with a single chromosome, a genome is interpreted as the same way as a replicon.
- a pair of genome $\mathcal{P}_G$:
- a set of genome $\mathcal{S}_G$:
- a set of data items $SET()$: a set of data items of any data type. $SET(\int)$ is equivalent to $\mathcal{S}_s$.

Once we have data types, we can define operations on each data type. Many existing bioinformatics tools can be interpreted as operations on genome data types. For example, biologists perform a sequence q comparison against a sequence database D, generating a set M of sequence matches in D. This analysis can be seen as BLAST$(q, D) \rightarrow M$. Biologists usually align a set S of sequences to extract possible common features using CLUSTALW, which can be interpreted as CLUSTALW(S). Furthermore, we can combine these two analyses. For example, biologists may want to align sequence matches with a certain score or higher from the BLAST search. By defining a wrapper, filterBLAST($\mathcal{S}$, cutoff), we can combine two analyses into one, CLUSTALW(filterBLAST(BLAST(q, D), cutoff)), which is a composition of several functions, BLAST, filterBLAST, and CLUSTALW. This composition is possible due to the data types and function definitions: BLAST: $\int \times \mathcal{S} \rightarrow \mathcal{S}$; filterBLAST: $\mathcal{S} \rightarrow \mathcal{S}$; and CLUSTALW: $\mathcal{S} \rightarrow \mathcal{A}$.

Data mining tools for genome comparison can be also defined on these data types as discussed in the next section.

5.2 High performance genome data mining tools

Comparing multiple genomes on the sequence level can be viewed as "integrating multiple sequence databases." Compared to the traditional data source such as texts and relational databases, biological sequences is unique in that there are no observable features in sequences, and relationship between sequences is obtained only via sequence analysis. Existing sequence analysis tools can be used to generate relationship among sequences or genomes. For example, two sequences $g_{i_k}$and g_{j_l}in two different genomes G_i and G_j can be

related if the two sequences share similarity of a certain level or higher. This sequence analysis based integration allows users to adjust the level of specificity, which is desirable since the interpretation of sequence analysis result is subjective. However, pairwise alignment tools generate only "binary" relationship between sequences. To draw more biological meaningful conclusion, it is necessary to combine a set of binary relationships to generate a set of sequences, often as a family of sequences, gene neighborhood, etc. This goal can be achieved more easily if there are data mining tools that are developed for comparative genome analysis tools. Below we describe a couple of examples. BAG is a sequence clustering algorithm which generates clusters of sequences based sequence similarity. Sequence clustering can then be interpreted as a function that takes a set of sequences and generates clusters of sequences, i.e., BAG: $\mathcal{S} \rightarrow SET(\mathcal{S})$. Another example is tools that compute gene neighborhoods, syntenic regions or segmental duplications in genomes. For example, FISH detects statistically significant segmental duplications from a genome pair matching data. This can be interpreted as FISH: $\mathcal{P}_G \rightarrow SET(\mathcal{P}_s)$.

More complicated tasks can be also specified using composition of functions. For example, PLATCOM provides MetaPATH, a service for comparing and investigating metabolic metabolic pathways in multiple genomes of users' choice as below:

(1) Genes involved in a certain metabolic pathway are retrieved from the KEGG metabolic pathway database either by selecting a reference genome and a certain metabolic pathway or by selecting a reference genome and inputting a query sequence in which case the user sequence is matched against genes in the selected genome using FASTA. This can be interpreted as FASTA($\int, \mathcal{G}$) where $\mathcal{G}$ is SET($\mathcal{C}$) and $\mathcal{C}$ is a set of genes involved in the pathway.

(2) The genes in the pathway are matched one by one against entire genomes that users selected, generating a gene-to-genome table that shows the presence or absence of genes in the pathway. This can be interpreted as **foreach** $g \in \mathcal{C}'$ and $\mathcal{G}_i \in SET(\mathcal{G})$ **do** FASTA($g, \mathcal{G}_i$)

(3) For any missing genes in the table, users can initiate a hidden Markov model (HMM) based search by clicking the `?` symbol to see whether the gene is truly missing or FASTA search simply could not detect an existing gene. A HMM is generated using genes in the same column as the missing gene and is used to search for the gene in the genome that corresponds to the row of the missing gene. This can be interpreted as HMMsearch($\mathcal{M}, \mathcal{G}$) where $\mathcal{M}$ is HMMbuild(SET($\int$)) and $\mathcal{G}$ is the genome corresponding to the row of the missing gene.

5.3 Genome analysis language

The genome analysis language for PLATCOM is at its infant stage. However, we have shown that a series of sequence analysis can be specified as a composition of functions. This is nothing new (many existing systems implicitly use the idea) but it is our intention to make this more formal based on function definitions. We create data types for genomes and import data mining tools that can be used for genome analysis, to enrich the syntax and semantics of the genome comparison language. Then, at any given point, a sequence analysis can be stored as a function composition together with the most recent result. In many cases, we can simply retrieve results from the intermediate results, rather than actually performing sequence analysis, which makes our argument of supporting interactive, exploratory analysis more appealing.

6 Conclusion

In this paper, we shared our experience in developing a genome comparison system PLATCOM and proposed a design paradigm for genome comparison systems.

PLATCOM allows users to choose genomes of their choice freely and perform analysis of the selected genomes with a suite of computational tools. PCDB is designed to incorporate new genomes automatically so that PLATCOM can evolve as new genomes become available. One important design feature of PLATCOM is use of high performance data mining tools to integrate separate system modules by gluing them together on the biological sequence level.

We proposed a design paradigm for PLATCOM that have been refined from our experience after the completion of the first implementation stage: defining genome data types, developing high performance data mining tools, and developing a genome analysis language.

References

[1] Walker, D.R, and Koonin, E.V. (1997), SEALS: A System for Easy Analysis of Lots of Sequences, *Intelligent Systems for Molecular Biology*, 5:333-339 Overbeek, R., Disz, T., and Stevens, R. (2004),

[2] Overbeek, R., Disz, T., and Stevens, R. (2004), The SEED: A Peer-to-Peer Environment for Genome Annotation, *Communications of the ACM*, 47, 47-50

[3] Dowel, R.D., Jokerst, R.M., Eddy, S.R., Stein, L. (2001), The Distributed Annotation System, *BMC Bioinformatics*, 2:7

[4] Uchiyama, I. (2003) MBGD: microbial genome database for comparative analysis, Nucleic Acids Research, 31, 58-62

[5] `http://amdec-bioinfo.cu-genome.org/html/caWorkBench.htm`

[6] Choi, K., Ma, Y., Choi, J.-H., and Kim, S. (2005), PLATCOM: A Platform for Computational Comparative Genomics, *Bioinformatics* (Application Notice) Advance Access published online on Feb. 24, 2005

[7] Choi, K. and Kim, S. (2005), CLASSEQ : Classification of Sequences via Comparative Analysis of Multiple Genomes, *Nucleic Acids Research* (under revision).

[8] Kim, S. (2003) Graph theoretic sequence clustering algorithms and their applications to genome comparison, Chapter 4 in *Computational Biology and Genome Informatics*, edited by Cathy H. Wu, Paul Wang, and Jason T. L. Wang, World Scientific.

[9] Calabrese, P., Chakravarty, S., and Vision, T.J. (2003), Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics*, 19, 74-80.

[10] Ma, Y., Bramley, R., and Kim, S., A Data Management Architecture for Computational Biology, Technical Report 607, 2005, Computer Science, Indiana University.

[11] Choi, J.-H., Cho, H.-G., and Kim, S., A Simple and Efficient Alignment Method for Microbial Whole Genomes Using Maximal Exact Match Filtering, *Computational Biology and Chemistry*, 2005 (submitted).

[12] Choi, J.-H., Choi, K., Cho, H.-G., and Kim, S. Multiple genome alignment by clustering pairwise matches. In Jens Lagergren, editor, *Proceedings of the 2nd RECOMB Comparative Genomics Satellite Workshop*, number 3388 in Lecture Notes in Bioinformatics, pages 30–41, Bertinoro, Italy, 2005. Springer-Verlag, Berlin.

[13] Choi, J.-H., Yang, J., and Kim, S., A Simple and Efficient Alignment Method for Microbial Whole Genomes Using Maximal Exact Match Filtering, *IEEE Computational Systems Bioinformatics Conference*, 2005 (submitted).

[14] Schwartz, S., Kent, W. J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R. C., Haussler, D., and Miller, W. (2003). Human-mouse alignments with BLASTZ. *Genome Res.*, 13(1):103–107.