Ancestral Genome Reconstruction in Bacteria

Kuan Yang

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Genetics, Bioinformatics, and Computational Biology

João C. Setubal, co-chair
Lenwood S. Heath, co-chair
Allan Dickerman
Boris A. Vinatzer
Brett Tyler

Jun. 6th, 2012
Blacksburg, VA

Keywords: phylogenetics, NGP, ancestral genome reconstruction, genome evolution simulation, homology, genomics

Copyright 2012, Kuan Yang

Ancestral Genome Reconstruction in Bacteria

Kuan Yang

ABSTRACT

The rapid accumulation of numerous sequenced genomes has provided a golden opportunity for ancestral state reconstruction studies, especially in the whole genome reconstruction area. However, most ancestral genome reconstruction methods developed so far only focus on gene or replicon sequences instead of whole genomes. They rely largely on either detailed modeling of evolutionary events or edit distance computation, both of which can be computationally prohibitive for large data sets. Hence, most of these methods can only be applied to a small number of features and species. In this dissertation, we describe the design, implementation, and evaluation of an ancestral genome reconstruction system (REGEN) for bacteria. It is the first bacterial genome reconstruction tool that focuses on ancestral state reconstruction at the genome scale instead of the gene scale. It not only reconstructs ancestral gene content and contiguous gene runs using either a maximum parsimony or a maximum likelihood criterion but also replicon structures of each ancestor. Based on the reconstructed genomes, it can infer all major events at both the gene scale, such as insertion, deletion, and translocation, and the replicon scale, such as replicon gain, loss, and merge. REGEN finishes by producing a visual representation of the entire evolutionary history of all genomes in the study. With a model-free reconstruction method at its core, the computational requirement for ancestral genome reconstruction is reduced sufficiently for the tool to be applied to large data sets with dozens of genomes and thousands of features. To achieve as accurate a

reconstruction as possible, we also develop a homologous gene family prediction tool for preprocessing. Furthermore, we build our in-house Prokaryote Genome Evolution simulator (PEGsim) for evaluation purposes. The homologous gene family prediction refinement module can refine homologous gene family predictions generated by third party *de novo* prediction programs by combining phylogeny and local gene synteny. We show that such refinement can be accomplished for up to 80% of homologous gene family predictions with ambiguity (mixed families). The genome evolution simulator, PEGsim, is the first random events based high level bacteria genome evolution simulator with models for all common evolutionary events at the gene, replicon, and genome scales. The concepts of conserved gene runs and horizontal gene transfer (HGT) are also built in. We show the validation of PEGsim itself and the evaluation of the last reconstruction component with simulated data produced by it. REGEN, REconstruction of GENomes, is an ancestral genome reconstruction tool based on the concept of neighboring gene pairs (NGPs). Although it does not cover the reconstruction of actual nucleotide sequences, it is capable of reconstructing gene content, contiguous genes runs, and replicon structure of each ancestor using either a maximum parsimony or a maximum likelihood criterion. Based on the reconstructed genomes, it can infer all major events at both the gene scale, such as insertion, deletion, and translocation, and the replicon scale, such as replicon gain, loss, and merge. REGEN finishes by producing a visual representation of the entire evolutionary history of all genomes in the study.

ACKNOWLEDGEMENT

I would like to thank Dr. Joao Setubal for his guidance during the years. I would like to thank Dr. Lenwood Heath especially for being my co-advisor and pushing me through the last stages of this dissertation. I would also like to thank all my other committee members, Drs. Allan Dickerman, Boris A. Vinatzer, Brett Tyler, for all their help and encouragement.

I would like to thank Dr. Ruth Grene for supporting me through the last year of my Ph.D and giving me the opportunity to work with the iPlant group.

I would also like to thank members in Dr. Joao Setubal's group, Andrew Warren, Chris Lasher, and Steven Mason for their interesting and good-spirited discussions relating to this research. Finally, thanks also go to Kelly Williams for making some scripts available.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

Ancestral genome reconstruction can be understood as a phylogenetic study of species of interest with more details than what is provided by a traditional phylogenetic tree. It may include information about ancestor species such as their gene content, the order of these genes in the genome, the replicon architecture, and the nucleotide sequence itself. Such information, when reliable, can help us better understand the evolutionary history of a set of organisms and thereby shed light on the genomic basis of phenotypes. Ancestral genome reconstruction is the topic of this dissertation.

Boussau *et al.* [1] reconstructed ancestral gene sets for a number of α-proteobacteria and quantified the flux of genes along the branches of the species tree. They inferred that the common ancestor of the α-proteobacteria was a free-living, aerobic, and motile bacterium with pili and surface proteins for host cell and environmental interactions. However, the authors inferred only the gene content of ancestral species. Slater *et al*. [2] inferred more detailed evolutionary histories for some members of the *Rhizobiales*, *Vibrionales*, and *Burkholderiales*. Their scenario for the *Rhizobiales*, based on extensive comparative genomic analysis, hypothesizes that the last common ancestral genome of members of this order had one chromosome and one plasmid. From this ancestor, several paths followed, some in the direction of enlarging this ancestral plasmid until it became a second chromosome (*Agrobacterium tumefaciens C58* and *Agrobacterium vitis S4*), and some in the direction of incorporating the plasmid into the chromosome (*M. loti, B. japonicum*), with other intermediary cases. However, Slater et al.'s reconstruction was

qualitative and did not provide a detailed reconstruction of gene content and genomic order. Moreover, their approach was not automated.

Automated methods for ancestral state reconstruction fall into two main categories, phylogeny based methods and genome rearrangement-based methods. Two of the best known phylogeny-based methods are the Sankoff algorithm [3] and the Fitch algorithm [4]. Although both algorithms are designed for ancestral nucleotide sequence inference, they can be adapted for gene order inference with slight modifications. For example, an ancestral gene reconstruction method based on neighboring gene pairs (NGPs) has been proposed [5]. An NGP is a pair of genes physically adjacent to each other on a replicon. Their method [5] extracts NGPs from genomes of extant species. The method then determines the occurrence of these NGPs in the ancestral genomes and outputs a list of conserved blocks assembled from the NGP content for each ancestor. The fundamental assumption of the method is that if adjacent homologous genetic loci are observed in both child species, then it is highly likely that they are also adjacent in the parent species. NGP-based methods can reconstruct ancestral genomes with thousands of genetic loci and have no limitation on allowed evolutionary events.

Compared to phylogeny based methods, genome-rearrangement-based methods usually start by simplifying genomes into strings of symbols, each of which represents a gene. Homologous genes are represented with the same symbol. No duplications are allowed in most of these methods and different heuristics are used to ensure it. This group of methods is extremely computationally intensive, as reconstructing a phylogeny from gene order data is NP-hard [6-8]. Although various heuristic methods have been developed [9],

they are still only applicable to small to medium sized data sets [5]. Furthermore, it has been suggested that this category of methods needs further study before they can yield reliable results in ancestral genome reconstruction [10, 11].

The rapid accumulation of numerous sequenced genomes demands detailed ancestral genome reconstruction methods that not only take into account all kinds of information but also are scalable to large genomes. Here we develop a computational system named REGEN (REconstruction of GENomes) for ancestral genome reconstruction. REGEN can cover most gene, replicon, and genome scale events, such as gene content reconstruction, contiguous gene run reconstruction, and replicon architecture reconstruction. However, it does not support nucleotide sequence reconstruction.

The performance of genome reconstruction at this scale relies heavily on the amount of available information, which consists of the genomes in the extant species represented as orthologs, a group of homologs.

Orthologs are related genes resulting from a speciation event in a single ancestral gene in the last common ancestor (LCA), while paralogs are genes that result from a duplication event [12]. The orthology concept is one of the cornerstones of genomics study, including gene function prediction [13]. Accurate orthology prediction is essential to any study in the comparative genomics field, including ancestral genome reconstruction, gene function annotation, gene function prediction by co-occurrence of genes [14] ,and even mutation effect prediction [15]. Much work has been done in this field, and many algorithms/software tools have been developed to identify orthologs [13, 16-24]. These methods have been categorized into three groups: tree-based methods, graph-based

methods and hybrid methods; a detailed review can be found in [25]. OrthoMCL [22] and InParanoid [17] are two of the most popular programs for ortholog identification and OrthoMCL is used in our study. A performance comparison of these tools has been done, but the results are inconsistent [26]. Despite their popularity, neither of these two methods uses information about local synteny during the ortholog prediction process. Moreover, they both can output both orthologous genes and paralogous genes. When such genes are present in a family, it means that we are unsure which ones are true orthologs of the other genes in the family. Because accurate orthology prediction is a key evolutionary technique for most comparative genomics studies [25] and only orthologs can be used in our reconstruction, we decided to undertake the problem of refinement of ortholog families using synteny information.

We have developed a systematic methodology to refine ortholog identification generated by third party *de novo* prediction programs. Our methodology targets the mixed families to obtain more orthologous families. Although gene synteny has already been used to confirm orthology prediction in prokaryotes [27], a formal methodology that combines synteny and phylogeny to refine orthologs prediction is still lacking. With an assumed reliable tree, this refinement method has essentially combined the strength of graph-based algorithms, phylogenetic information, and local synteny in the ortholog identification process. We show in Chapter 4 that our refinement methods can successfully turn almost 80% of the mixed families produced by orthoMCL into orthologous families with $p < 0.05$.

Another important area involved in ancestral genome reconstruction is validation. Simulated data are usually employed for this purpose due to the nature of the study. To this purpose, we developed a random events-based prokaryote genome evolution simulator that we call Prokaryote Evolutionary Genomics Simulator (PEGsim) [28]. It is capable of simulating medium- to large-scale evolutionary events, an area in which good simulators are lacking. Species-, replicon-, and gene-level events, such as speciation, replicon fission and fusion, replicon gain and loss, replicon merge and split, gene gain and loss, gene transposition and translocation, gene duplication and reversal, and horizontal gene transfer are implemented in PEGsim. PEGsim also implements the concept of conserved gene runs, which can be mapped to the biological concept of operon.

An important principle in the design of PEGsim is simplicity and efficiency of use. The program runs in linear time the total number of genes in the entire group of species. Running time can vary depending on parameter settings. However, a simulation with one starting chromosome of 3000 genes and a plasmid of 1000 genes and resulting in 10-15 extant species with reasonable settings of other parameters finishes within minutes. To our knowledge, existing evolution simulation tools that are comparable to PEGsim are dawg [29], evolsimulator [30], and GSIMULATOR [31]. However, they all lack models for gene- and replicon-scale evolutionary events. PEGsim is the first simulator designed to fill these gaps. It has probability-based models for all general gene- and replicon-scale events in prokaryotes. PEGsim is described in Chapter 5.

With both of the previous components developed, we are able to perform accurate ancestral genome reconstruction with REGEN and provide measures of confidence in regard to obtained results. Furthermore, we also evaluated its performance by comparing with previous studies.

There are a few things we can try to improve REGEN. First of all, REGEN is built based on NGPs, which are dimers of genes. A reasonable extension is to increase the number of genes to three so the reconstruction is carried out on trimers. Second, general graph algorithms are used in the gene run reconstruction process and take the majority of the running time consumed by REGEN. More refined algorithms designed with the consideration of the nature of gene run graphs should reduce the running time substantially.

Our work is the first to perform model-free NGP-based ancestral genome reconstruction in a fully automated fashion, while supporting both maximum parsimony and maximum likelihood criteria. We apply REGEN to a group of *Rhizobiales* species that vary significantly in life styles (e.g., plant pathogens, animal pathogens, mutualists, and free-living bacteria), genome architecture (e.g., single chromosome, pair of chromosomes, with and without plasmids, and large and small plasmids), and genome size.

In the remainder of the dissertation, we first define some of the important concepts involved in this work (Chapter 2). Then, we provide a formal definition of the targeted problem (Chapter 3). Finally, we show the details for development and application of each component involved in the reconstruction, including homology refinement (Chapter 4), PEGsim (Chapter 5), and REGEN (Chapter 6).

Chapter 2

Concepts and Definitions

Our model of a prokaryotic genome is that it contains a main chromosome and zero or more additional replicons. These other replicons can be additional chromosomes and/or plasmids. It is worth pointing out that additional chromosomes in prokaryotes are the exception and not the norm, at least in species whose genomes have been sequenced. Some important concepts used in this dissertation are listed below.

Speciation: the splitting of lineages. One ancestor splits into two child species. It can happen at most once per generation for each species.

Replicon: a self-replicating DNA unit in a genome, such as a chromosome or a plasmid.

Replicon merge: Two replicons are merged into a new replicon. If either of the original replicons is the main chromosome, then the new replicon remains the main chromosome, otherwise a new name is created. It can happen at most once per generation for each replicon.

Replicon split: A single replicon is split into two new replicons. If the original replicon is the main chromosome, then the larger of the two new replicons will be named the main chromosome. It can happen at most once per generation for each replicon.

Replicon loss: A replicon is lost. Main chromosome cannot be lost. It takes place at most once per generation for each replicon.

Replicon gain: A species gains a new replicon. It takes place at most once per generation for each species.

Gene gain: A single gene or consecutive run of genes is gained. It takes place at most once per position per generation for each replicon of every species.

Gene loss: A single gene or consecutive run of genes is lost. It takes place at most once per position per generation for each replicon of every species.

Gene reversal: A single gene or consecutive run of genes is reversed. It takes place at most once per position per generation for each replicon of every species.

Gene duplication: A single gene or consecutive run of genes is duplicated and inserted into a random location on the same replicon.

Gene translocation: A single gene or consecutive run of genes is transferred from one replicon to another in the same species.

Gene transposition: A single gene or consecutive run of genes is transferred from one position to another on the same replicon.

Horizontal gene transfer: A single gene or consecutive run of gene is transferred from one species to another that is evolving at the same time.

Homologous gene family: A group of structural and/or functional similar genes descended from the same ancestor.

Orthologous gene family: A homologous gene family with only orthologous genes, namely one gene in each species. Each orthologous gene family is assigned a unique ID.

Paralogous gene family: A homologous gene family with all genes coming from one single taxon.

Mixed gene family: A homologous gene family with orthologous genes from some species and paralogous genes from other species.

Core gene: A gene that occurs on the main chromosome of all the species in a study.

Singleton gene: A gene that does not have a homologous counterpart in any other genome in the study and is represented by '*' in the reconstruction.

Gene family alphabet $\sum$: the set of all orthologous gene family IDs plus *.

Gene run: A chain of genes located consecutively on a replicon represented by a finite sequence over $\sum$ from a genome without interruption by *.

Conserved blocks: A conserved gene run across a group of species. Conserved blocks will be affected by evolutionary events much more rarely than other blocks.

Phylogenomic tree: a species tree built based on concatenated aligned protein sequences of thousands of genes appearing exactly once in all genomes of interest.

Neighbor Gene Pair (NGP): a pair of genes physically adjacent to each other on a replicon.

Chapter 3

Problem Statement

Given a group of complete genomes from closely related bacterial species (usually species in the same order, such as family, or genus) and a rooted species tree of these genomes (which define their phylogenetic relationships), infer gene set, gene order, and replicon architecture for each internal node in the tree.

Note that nucleotide-scale evolutionary reconstruction is not considered in this project. Models for nucleotide evolution form their own research area and have been intensively studied [32-35].

Input

1. Complete annotated genomes of a group of bacteria of interest and of a certain number of outgroup genomes. Only protein-coding genes are included. For each gene, the following information is required: GeneBank accession number or ID, product, strand, and genome coordinates.

2. Orthologous gene families across these genomes. This information can be obtained by running an ortholog family computation program. In this work, we have used for this purpose the program OrthoMCL [22].

3. A trusted rooted phylogenetic tree of the input species.

Output

For each ancestral genome (internal node in the input species tree):

1. A hypothesis about genome architecture (number of replicons, type of replicons).
2. The overall set of genes.
3. Relative location of each gene to each other and strand information of each gene when possible.
4. Replicon assignment for each gene when possible.
5. Annotation of the tree branches with genome-wide evolutionary events, including reversal, translocation, replicon acquisition/loss, replicon split, duplication, and lateral gene transfer.

Chapter 4

Homology Prediction Refinement

Homology designates the relationship of entities that share a common ancestor, regardless to the possible evolutionary events that led to the current situation [36]. Genes that share such a relationship are referred to as homologs. Homologs can be further classsified into two groups, orthologs and paralogs. Two genes are said to be orthologs when the evolutionary event that gave rise to them was a speciation event. Two genes are said to be paralogs when the evolutionary event that gave rise to them was a duplication event.

OrthoMCL [22] and InParanoid [17] are two of the most popular programs for ortholog identification. Here we describe the development of a systematic methodology to refine the ortholog identification generated by such programs. We also show the improvement made by the refinement using a pilot reconstruction on a small group of *Rhizobiales* species with and without homology refinement.

4.1 System and Methods

The refinement method assumes a reliable species tree, to be provided as input. It also assumes results from a *de novo* homologous gene family prediction program, such as OrthoMCL.

We consider each genome to be a collection of replicons. In the *Rhizobiales* order, most fully sequenced genomes have more than one replicon (e.g., one chromosome and several plasmids).

4.1.1 Genome preprocessing

Each replicon is represented by an array, with each array element representing a gene present in the replicon. When a gene belongs to an orthologous family, the family ID is used to represent the gene; when the gene does not belong to any family the symbol '*' is used. The process ends with a list of replicons containing only orthologous gene family IDs.

4.1.2 Homology refinement

The refinement consists of four main steps: building profiles for the extant species, profile comparison and merging, ortholog family assignment, and statistical confidence assessment.

4.1.3 Profile building for extant species

The refinement is carried out in a family-by-family fashion. For each member of the family, we take 10 genes upstream and downstream to form a profile for this gene. The gene order is ignored during profile building to simplify the profile merging process. For each of these ten genes, the profile contains orientation and replicon location information with an initial weight. Currently, we only distinguish whether a gene is on the main chromosome or on a plasmid. Genes on different plasmids are considered to be in the

same replicon location. We compensated for this bias by assigning lower weight when two genes are both on plasmids instead of chromosomes. A species with two paralogous genes in the same family will have two profiles, one profile for each gene, after this process. A simple example is shown in Figure 4.1.

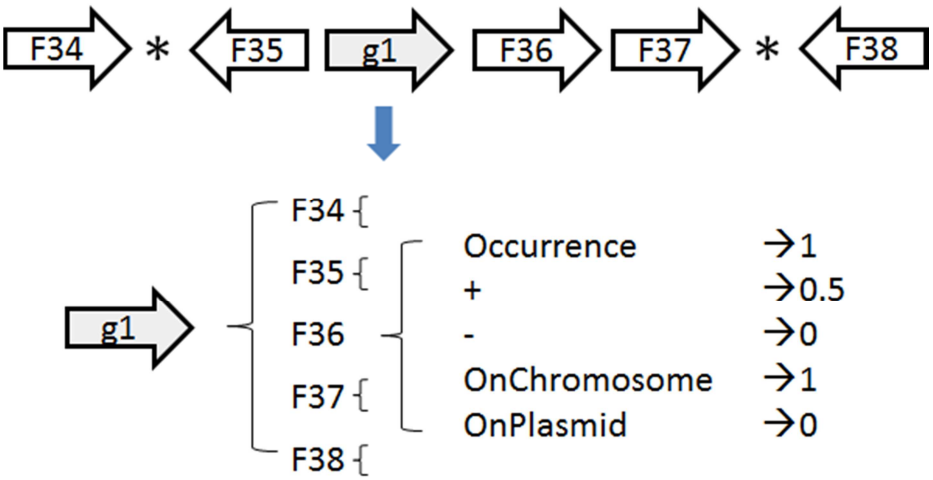

Figure 4.1. The profile built for the gene *g1*. It includes *occurrence, strand, and localization (either on a chromosome or plasmid) of genes around g1. In this particular example, the profile shows the status of five genes, F34, F35, F36, F37, and F38. Occurrence is set to 1 to simply show F36's occurrence. '+' is assigned 0.5 because the gene is on the plus strand. The gene run is on chromosome so onChromosome is also set to 1. All values are default and can be customized.*

4.1.4 Profile comparison and merging

Using the input species tree, we will compare the profiles and merge them when appropriate. This process proceeds in a bottom-up fashion, namely it starts from the most recent ancestors of two leaves in the tree and finishes at the root of the tree. The level of

each ancestor (how many speciation events away from the extant species) is determined by a simple traversal of the tree. From the lowest level, for each ancestor in that level, we identify its children and perform an all-by-all comparison for all the profiles contained in both children. In each comparison of two profiles, we check the number of shared orthologous genes, whether they are in the same orientation and located on the same replicon or not. A comparison score for each profile pair is calculated based on these three criteria and stored in a list. The profile pair that achieves the highest score will be merged and removed from the list. This merging process is repeated until the highest remaining score drops to 0. Any unmerged profiles from children are directly assigned to the ancestor as well. Each profile can only be merged once. If it is already merged, the profile pair is simply ignored and removed. The following pseudocode gives a better view of the entire process.

Pseudocode

1. Determine levels (L)of the tree

2. ***for*** *each* $l \epsilon L$

3. ***do*** *ANC* $\leftarrow$ *ancestors in l*

4. ***for*** *each* $a \epsilon$ *ANC*

5. ***do*** *identify offspring A and B*

6. $P_A \leftarrow$ *profiles in A*

7. $P_B \leftarrow$ *profiles in B*

8. ***for*** *each* $j \epsilon P_A$

9. ***for*** *each* $k \epsilon P_B$

10. *do compute score*

11. *sort the scores in descending order(S)*

12. ***for*** *each s* ϵ *S*

13. ***if*** *s* $\neq$ *0 and contributing profiles(ps) still available*

14. ***do*** *pn* $\leftarrow$ *merge the ps*

15. *mark ps unavailable*

16. *Assign pn to a*

17. *Assign all remaining unmerged profiles to a*

During the merging process, the scores for the shared orthologous families are summed to increase their weight in the profile. Orthologous families that do not have a match in the other profile are also kept in the new profile, with the consideration that they may match other profiles in future comparisons. A simple example is shown in Figure 4.2.

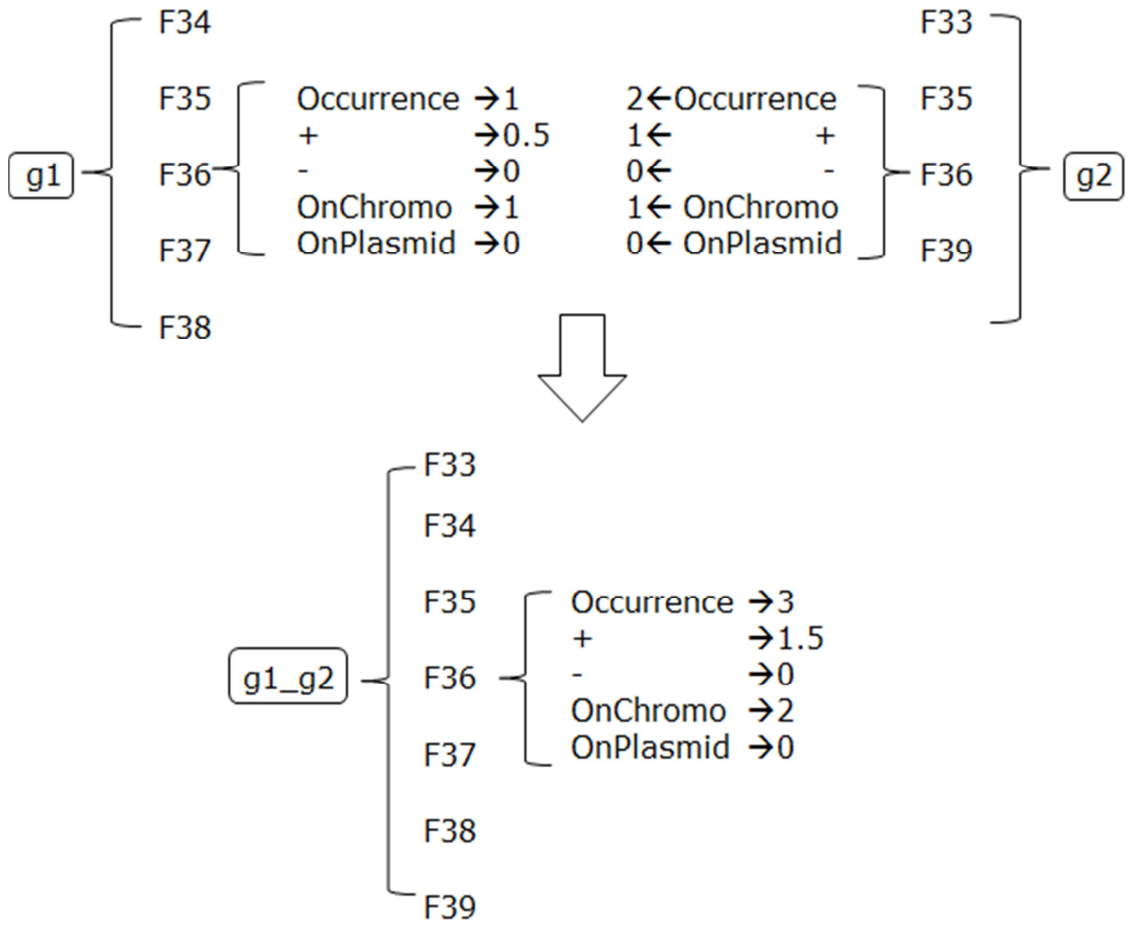

Figure 4.2. Profile comparison and merging. *For simplicity only the comparison and merging process for F36 are shown. The process is carried out for all matched orthologous families, such as F35 in this case. In the merging process, corresponding values are summed. For example, the Occurrence value for F36 in the profile of g1_g2 is the sum of occurrence values in the previous two profiles for g1 and g2.*

4.1.5 Orthologous family assignment

When the comparison and merging process is finished at the root of the tree, the genes in each of the profiles at the root are assigned to a new orthologous gene family. Since gene synteny alone is not a sufficient criterion for accurate orthology prediction, we respect the

original decision made by the previous software by trying to merge these new orthologous families, as long as the merging process does not create paralogs. We show an example to clarify this issue. Suppose that in a homologous family identified by OrthoMCL there are the following genes: g_1 and g_2 from species *A*, g_3 and g_4 from species *B*, and finally g_5 from species *C*. Suppose further that our method determines that g_1 and g_3 should be in an orthologous family (F_1) and g_2 and g_4 should be in another orthologous family (F_2), and g_5 is left out of both new families. In this case, we will try to put g_5 back in either F_1 or F_2. The priority for an orthologous gene family to incorporate other genes increases with its size, namely the number of members already in the family. In the cases where several paralogs from the same species can be added to the same newly identified family, the one with the lowest e-value by BLASTP [37] with any existing member of the family is added (all-against-all BLASTP e-values are available as inputs to orthoMCL).

4.1.6 Statistical confidence assessment

We used a randomization approach to assess the confidence for each refined orthologous gene family. To make the gene order as similar to the real data set as possible, we only randomized the order of genes on each replicon. The number of species and their replicon configuration are not changed. The confidence of each family is reflected by summation of all scores in the final profile at the root. The percentage distribution of the scores in the real dataset and the one from the randomized dataset is shown in Figure 4.3. The randomization is run for 100 times and the average is given as the result. Over 92% of the

identified orthologous gene families are statistically confident at a p-value of 0.05. All statistically insignificant families are removed from further analysis.

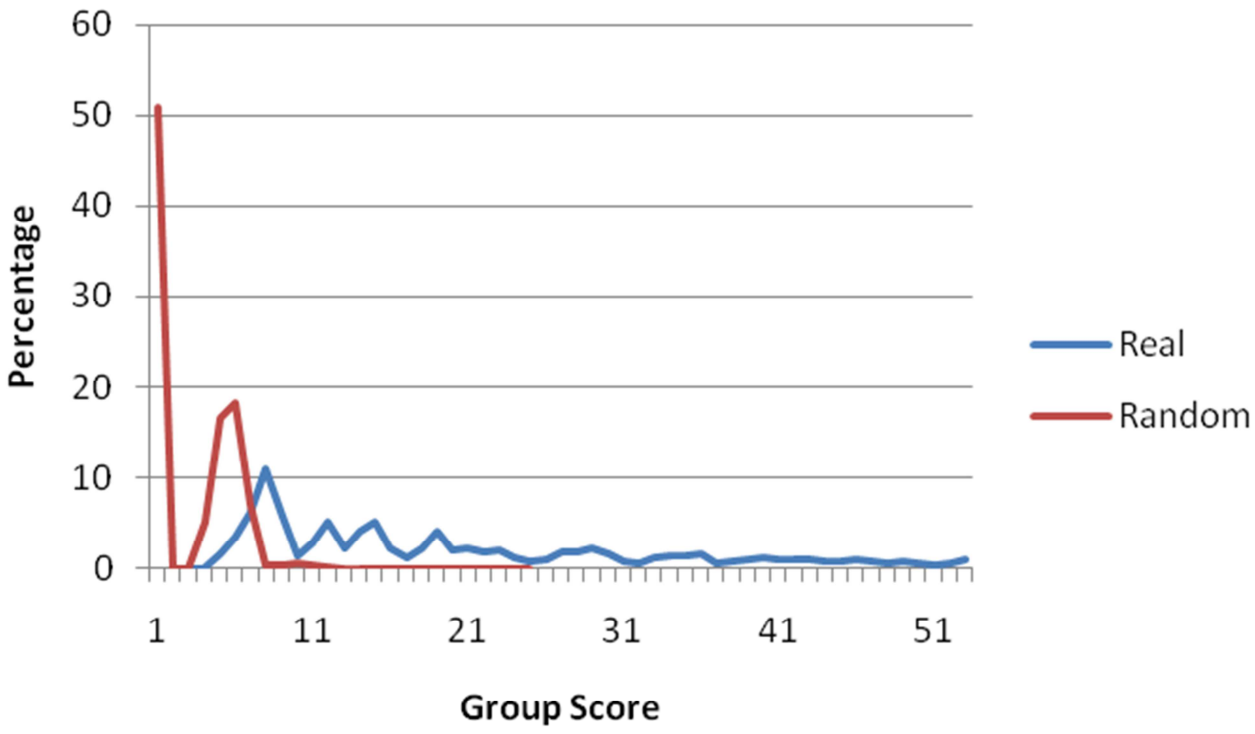

Figure 4.3. Distribution of group scores in real and randomized dataset.

4.2 Results and Discussion

We applied our method to a collection of 10 Rhizobiales species: *Agrobacterium radiobacter K84*, *Agrobacterium tumefaciens str. C58*, *Agrobacterium vitis S4*, *Bradyrhizobium japonicum USDA 110*, *Brucella suis 1330*, *Mesorhizobium sp. BNC1*, *Mesorhizobium loti MAFF303099*, *Rhizobium etli CFN 42*, *Rhizobium leguminosarum bv. viciae*, *Sinorhizobium meliloti 1021*, with *Bradyrhizobium japonicum USDA 110* as

the outgroup to the other nine Rhizobiales species. Genome sequences were downloaded from Genbank database release 176.0. OrthoMCL 1.4 was used to identify homologous families. The species tree that we used to test our methodology was the one presented by Slater et al. [2], which was obtained by a super matrix approach [38]. This tree was generated using 423 orthologous sequences.

In total, OrthoMCL returned 9,237 homologous families: 6,939 orthologous families with 34,643 genes, 1,698 mixed families with 13,047 genes, and 600 single species paralog families with 1,530 genes. Single species paralog families were not taken into consideration for this study. The refinement method identified 1764 orthologous families with p-value = 0.05. Detailed information is shown in Table 4.1.

Table 4.1. Result of the refinement process and related information.

	OrthoMCL		Refinement		Used for reconstruction
	C1	C2	I	O	
F	6,939	1,698	1,320	1,764	8,703
G	34,643	13,047	11,320	9,150	41,354

**C1: orthologous gene family; C2: mixed homologous gene family; I: mixed homologous gene families that can be processed by the refinement method, some of the mixed families cannot be processed; O: output from the refinement; F: number of gene families; G: number of genes;*

In Table 4.1, the OrthoMCL column specifies the numbers of orthologous families and mixed families identified by orthoMCL, as well as the total number of genes included in all these families. The refinement column shows how many of the mixed families could

be processed by the refinement method and how many valid orthologous families are identified (the refinement column). The used-for-reconstruction column shows the merged orthologous gene families from both OrthoMCL and the refinement, which serves as input for the next step, genome reconstruction.

As shown by Table 4.1, ~77% of all the mixed families (87% of all genes in these families) can be refined by our method and resulted in 1,764 orthologous families (70% of all genes in mixed families). It increased the number of orthologous families and genes used for reconstruction by ~25%, from 6939 to 8703.

This kind of improvement significantly helps the reconstruction of ancestral conserved syntenic blocks. We have successfully reconstructed 439 conserved blocks for the last common ancestor of nine *Rhizobiales* species with *Bradyrhizobium japonicum USDA 110* as outgroup. This number drops to 393 when refined orthologous families are not used in the reconstruction. More importantly, including refined orthologous families increased the size of the longest conserved blocks from 22 genes to 31 genes. The distribution of lengths of conserved blocks with the minimum size of 8 genes from both datasets is shown in Figure 4.4. Conserved block reconstruction was carried out for all internal nodes in the species tree, and we obtained conserved blocks for all ancestral genomes.

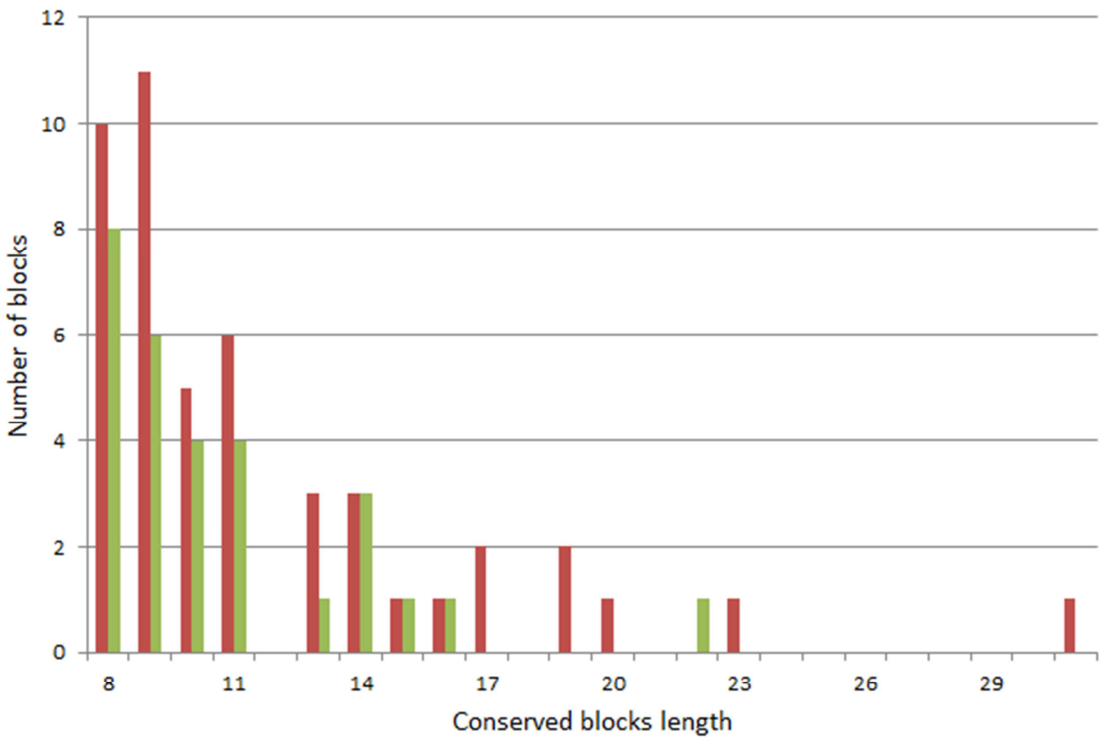

Figure 4.4. Distribution of blocks reconstructed for the last common ancestor. *The red bars are from the reconstruction with refined orthologous families and the green bars without refined orthologous families.*

It is easy to see that the refinement not only helps to reconstruct more conserved blocks, but also increases the length of the conserved blocks. The longest reconstructed conserved blocks contains 31 genes with refined orthologous families, in constrast to 22 genes without refined orthologous families. This result is at least comparable to the ones reported in [5] for the eukaryote *Drosophila*, considering the difference in the number of orthologous genes among the Rhizobiales here considered and the *Drosophila* species considered in [5]. This shows that it is possible to undertake genome reconstruction in

prokaryotes, at least for groups of genomes that are close enough, such as those from the Rhizobiales that we have used.

4.3 Additional Remarks

We have formalized a systematic methodology to refine ortholog identification predictions generated by third party *de novo* prediction programs by combining local synteny and phylogeny. More than three quarters of all the mixed homologous gene families can be processed by this method, and 92% of the newly identified orthologous gene families are statistically significant at a p-value of 0.05. These numbers are expected to grow when the method is applied to eukaryotic genomes.

This is the first computational method that can systematically refine the result from other *de novo* orthology identification programs with statistical support by combining local synteny and phylogeny. It is also the first method that can reconstruct conserved blocks for ancestral genomes with fully resolved strand information in bacteria. However, there are several important assumptions and simplifications made by the program. First of all, the entire reconstruction algorithm is a maximum parsimony based method, which has proven to be less accurate as the branch length increases. The parsimony criterion assumes that the presence of more events to explain the same present-day situation is a less likely occurrence than one that uses fewer events to explain the same situation. It also assumes that after a gene duplication event, the gene that remains at the original location will retain its original function and hence is the functional ortholog of the family. This might be true in most cases, but not in all of them. A true functional ortholog among

paralogs can only be determined by wet-lab experiments and/or gene expression data. On the other hand, these very simplifications make this method possible and make the reconstruction taking ~8500 ortholog families from ten species practical. For the Rhizobiales genomes, the refinement took less than 1 minute and the reconstruction took less than 10 minutes on a standard desktop computer.

A version of this chapter was published as Yang K, Setubal JC: Homology prediction refinement and reconstruction of gene content and order of ancestral bacterial genomes. In: *Proceedings of the 2010 ACM International Conference on Bioinformatics and Computational Biology*: 2010; Niagara Falls, New York, U.S.A (full paper) [39].

Chapter 5

A Whole Genome Simulator of Prokaryote Genome Evolution

Here we present the development and performance evaluation of PEGSim, a random events-based genome evolution simulator. The goal of PEGsim is to simulate medium- to large-scale evolutionary events, species-, replicon-, and gene-level events, such as speciation, replicon fission and fusion, replicon gain and loss, replicon merge and split, gene gain and loss, gene transposition and translocation, gene duplication and reversal, and horizontal gene transfer (see definitions of these concepts in Chapter 2). Nucleotide sequence scale events, such as substitution, are not included.

Parameter setting has always been a challenge in the development of simulation tools. We derived some of the default parameter values in PEGsim from a recent extensive survey of prokaryotic genome evolution [40].

5.1 System and Methods

The following subsections describe the features implemented in PEGsim and the underlying model.

5.1.1 Defining the genome of the last common ancestor (LCA)

The genome of the LCA can have any number of chromosomes and plasmids of any size. Genes are represented by numbers and orientation by +/-. The distribution of genes on the

plus strand and the minus strand is customizable. 70% of the genes are assigned to the plus strand in this simulation.

5.1.2 The global simulation

Users can define the number of generations and the base rate of each evolutionary event. The actual evolutionary rate will fluctuate from one species to another. Each species can have at most one species-scale event in a single generation, each replicon one replicon-scale event and each position on a replicon one gene-scale event. The strength of conservation for conserved blocks can also be customized, with a default value set at 0.8, meaning 80 percent of the gene level events that are supposed to happen at a conserved position are rejected. Chromosomes and plasmids can have different values for the strength of conservation. End users can also decide which scale events shall occur in the simulation. We found this feature very useful when the study is focused on a specific kind of event. All events are allowed to occur by default.

The species that are evolving at any generation are also recorded. This list is used to determine the source and target species for horizontal gene transfers, which can only happen between species that are evolving at the same time.

Two separate streams of pseudorandom numbers are used in PEGsim, one to control the speciation events and the other the rest of the evolutionary events. The number of genes involved in each evolutionary event is also recorded in the order of occurrence. Given this information, PEGsim can repeat any simulations when desired, or reproduce a given phylogenetic topology with different sets of gene- and replicon-scale events.

5.1.3 Probabilistic model for evolutionary events

In the beginning of each generation, every evolving species has a certain probability to have a speciation event. If a speciation event takes place, two child species with identical genomes as the parent species are born, and the parent is eliminated from the currently evolving species list.

Each replicon in each species has a certain probability to have at most one replicon-scale event, such as a replicon split, at each generation. For example, if a replicon has already gone through a replicon split event, then it will not have any other event for the generation. Chromosomes have a lower evolutionary rate than other replicons by default.

Each position in a replicon has a certain probability to have at most one gene-scale event, such as a gene reversal. For example, if a position already had a gene reversal, it will not have any other gene-scale event for the generation.

A species can have at most one event from any scale for any generation. The priority for the events is: species-scale, replicon-scale, and gene-scale. For example, if a species goes through a speciation event, it will not be considered for any other events. If a species does not go through a speciation event but does go through a replicon-scale event, it will not be considered for any gene-scale event.

5.1.4 Gene Content

The gene content change in PEGsim is achieved by gene birth and gene loss. In PEGsim, gene birth can be achieved through different events, including gene duplication,

endogenous HGT, and exogenous HGT. Gene duplication here means that after the duplication, the duplicated genes still have the original function, so they are represented by the same gene id. Endogenous HGT can transfer a group of genes from one species to another, thus changing the gene content of the target species. Exogenous HGT can happen at any position with some genes that have never been seen in the group of species before, i.e., these genes come from a donor species that is not part of the simulation.

5.1.5 Power law distribution based length

All gene-scale events in PEGsim can involve one to several genes, with the exact number determined by a power-law distribution. Such distributions are ubiquitous in both natural and artificial phenomena, including physics, biology, geography, and even Internet ecology [41, 42]. To fit the real world, the number of genes involved in each event is drawn from customizable power-law distributions. The power-law number generator used in the simulator follows the following formula:

$$X = [(a^{n+1} - b^{n+1}) \times y + b^{n+1}]^{1/(n+1)}$$

where a and b are the maximum and minimum values of the distribution, respectively, y is a uniformly distributed variable on [0,1], and n is a constant that affects the shape of the curve [43, 44].

5.1.6 Conserved blocks

Conserved blocks are created when the genome is initialized for the LCA of all the species in the simulation. The percentage of a replicon covered by conserved blocks and

the length of the conserved blocks are both customizable. By default, 15% of the main chromosome and 5% of any plasmid are covered by conserved blocks. The lengths of the conserved blocks are also drawn from power-law distributions.

The preservation of conserved blocks is achieved in two ways. First, any mutation scheduled to happen to genes located in conserved blocks could be rejected according to a certain probability (mentioned above). Second, for the events that have passed the first rule, they could be adjusted to affect the entire conserved block instead of breaking it. For example, if a part of a conserved block is to be reversed, the event could be replaced by a reversal of the entire conserved block. The criteria for both methods are customizable.

5.1.7 Simulation flow overview

The following pseudocode gives a general overview of the method and explains the priority of different events. Every step in the algorithm also includes necessary updates of related information, such as the list of evolving species and the conserved blocks. Every simulated event takes place following its own rate or distribution; this is implicit in the conditional statements below of the form "if <event> takes place".

Pseudocode[1]

1. LCA genome initialization with conserved blocks

*2.**for** generation (2..end)*

*3. **for** each species that is evolving*

*4. **if** speciation takes place*

[1] rs = replicon split; rl = replicon loss; rm = replicon merge; ra = replicon acquisition; gl = gene loss; gi = gene insertion; gtr = gene transfer; gtl = gene translocation; gr = gene reversal;

5. ***do*** *speciation and end loop*

6. ***for*** *each replicon the species has*

7. ***if*** *rs takes place*

8. ***do*** *rs and end loop*

9. ***if*** *rl takes place*

10. ***do*** *rl and end loop*

11. ***if*** *rm takes place*

12. ***do*** *rm and loop*

13. ***if*** *ra takes place*

14. ***do*** *ra and end loop*

15. ***for*** *each position on the replicon*

16. ***if*** *gl takes place*

17. ***do*** *gl and move to the next available position*

18. ***if*** *gi takes place*

19. ***do*** *gi and move to the next available position*

20. ***if*** *gtl takes place*

21. ***do*** *gtl and move to the next available position*

22. ***if*** *gtr takes place*

23. ***do*** *gtr and move to the next available position*

24. ***if*** *gr takes place*

25. ***do*** *gr and move to the next available position*

26. ***if*** *HGT takes place*

27. ***do*** *HGT and move to the next available position*

28. ***return*** *genomes and tree*

5.1.7 Output

After a simulation is completed, PEGsim provides the output described in the next four paragraphs.

Genomes of the extant species and the ancestors

The genomes of all species that ever existed in the simulation are recorded. The file is in a FASTA-like format, where, instead of DNA sequence, we list the gene IDs with plus (+) or minus (–) signs representing orientations.

A phylogenetic tree

A phylogenetic tree that describes the evolutionary history of all the species involved in the simulation is output in the NEWICK format.

Conserved blocks

This file contains the conserved blocks for all species. For ancestors, it contains the conserved blocks when the ancestor encountered a speciation event and stopped evolving. For extant species, it contains the conserved blocks when the simulation ends.

Events log

The output also includes a detailed log of every event that happened during the simulation in the order of occurrence. This log is the true history of the simulated evolution and can be used to benchmark different analytic methods and algorithms.

5.1.8 Simulator behavior evaluation

We generated a large amount of simulated data to observe the behavior of the simulator. A group of 22 fully sequenced species from the order *Rhizobiales* in the *alpha-proteobacteria* were selected to compare to simulated data in a number of situations to guide the parameter setting. The *Rhizobiales* order is known to have bacteria with very different lifestyles (plant pathogens, animal pathogens, free living mutualists), varying genome architectures (single chromosome, pair of chromosomes, with and without plasmids, and large and small plasmids), and a large range of genome sizes (from 1 Mb to 9 Mb). The selected species are *Agrobacterium radiobacter K84, Agrobacterium tumefaciens C58 Cereon, Agrobacterium vitis S4, Azorhizobium caulinodans ORS 571, Azospirillum B510 uid32551, Bartonella henselae Houston-1, Beijerinckia indica ATCC_9039, Bradyrhizobium japonicum, Brucella suis 1330, Hyphomicrobium denitrificans ATCC_51888_uid33261, Mesorhizobium BNC1, Methylobacterium chloromethanicum CM4, Methylocella silvestris BL2, Nitrobacter hamburgensis X14, Ochrobactrum anthropi ATCC_49188, Oligotropha carboxidovorans OM5, Parvibaculum lavamentivorans DS-1, Rhizobium etli CFN_42, Rhodomicrobium vannielii ATCC_17100_uid38253, Rhodopseudomonas palustris BisA53, Sinorhizobium meliloti, Starkeya novella DSM_506_uid37659,* and *Xanthobacter autotrophicus Py2.*

Since duplicated genes in the real data set are eliminated using the method described in [39], we disabled gene duplication in all the simulations shown below. The LCA of all the following simulations has a chromosome of 3000 genes and a plasmid with 500 genes, and the simulations are set to run for 3000 generations. The power-law variant generators used in the simulation are customized as shown in Table 5.1. The constant *n* is set to 20 in all the generators.

Table 5.1. Properties of the power-law number generators used.

Event	**Max (a)**	**Min (*b*)**
Insertion	50	1
Loss	50	1
Transposition	50	1
Translocation	50	1
Reversal	500	1
HGT	50	1
Conserved Blocks	40	2

Genome Size evaluation

The genome size of the simulated genomes can be regulated by adjusting any one of or combinations of the following parameters: gene loss rate, gene insert rate, and horizontal gene transfer rate. As expected, a high gene insertion rate will increase the genome sizes while a high loss rate will decrease them. Increasing both parameters leads to larger differences among the size of the genomes. By altering the parameters, we were able to

obtain simulations with similar mean and standard deviation as the real data. We randomly selected ten such simulations and plotted them with the real data in Figure 5.1. The real data is shown in red.

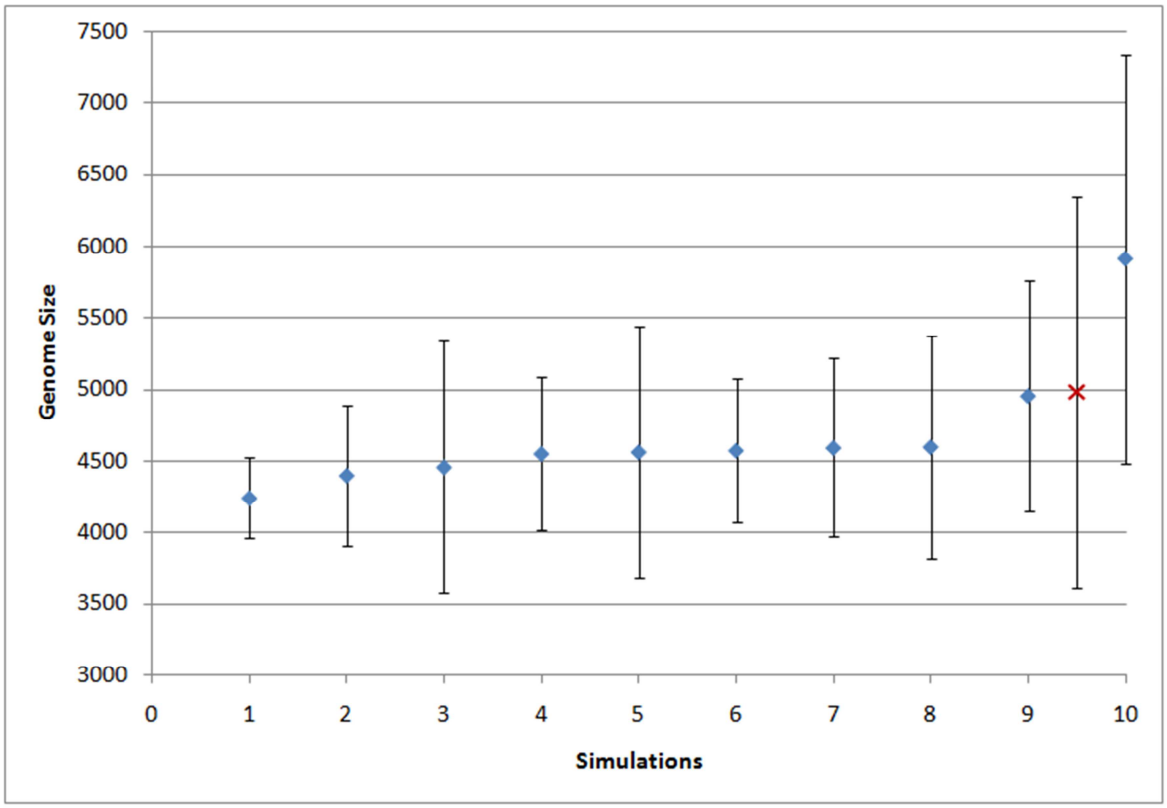

Figure 5.1. Genome size mean and standard deviation of ten simulations and the *Rhizobiales* data set (red cross on the right). **Gene loss, gene insertion and HGT rate are 9e-5, 8e-5, and 6e-5 respectively.*

This result shows that PEGsim can generate simulated data sets with properties similar to real genomes. Users can modify the related parameters to generate simulations according to their needs.

As can be observed, the simulated data sets have a slightly smaller standard deviation compared to the real data set. We believe the reason for this difference is the random events-based model used in PEGsim. Genome evolution is not a random process [45, 46], and genome expansions or contractions can be triggered by the change of environment. These changes can be highly directional. For example, when a certain bacterial species moves into a new environment with different nutrient sources, it may have to pick up a large number of genes through HGT to survive, thus a rapid genome expansion is expected. An opposite scenario would be free-living bacteria that adopt an intracelullar lifestyle, so that a significant part of their genes are not needed anymore, leading to genome contraction [1]. As we have pointed out earlier, the order *Rhizobiales* contains bacteria with very different life styles and genome architectures, leading to a large standard deviation. In order to demonstrate PEGsim's capacity to generate more conserved simulated data, we compared another group of simulations with the available genomes of the *Brucella* genus in the Genbank database. The genomes used are *Brucella abortus bv. 1 str.9-941 (uid58019),Brucella abortus S19 (uid58873),Brucella canis ATCC 23365 (uid59009), Brucella melitensis ATCC 23457 (uid59241), Brucella melitensis biovar Abortus 2308 (uid62937), Brucella melitensis bv. 1 str. 16M (uid57735), Brucella microti CCM 4915 (uid59319), Brucella ovis ATCC 25840 (uid58113), Brucella suis 1330 (uid57927), and Brucella suis ATCC 23445 (uid59015).* The result is shown in the Figure 5.2.

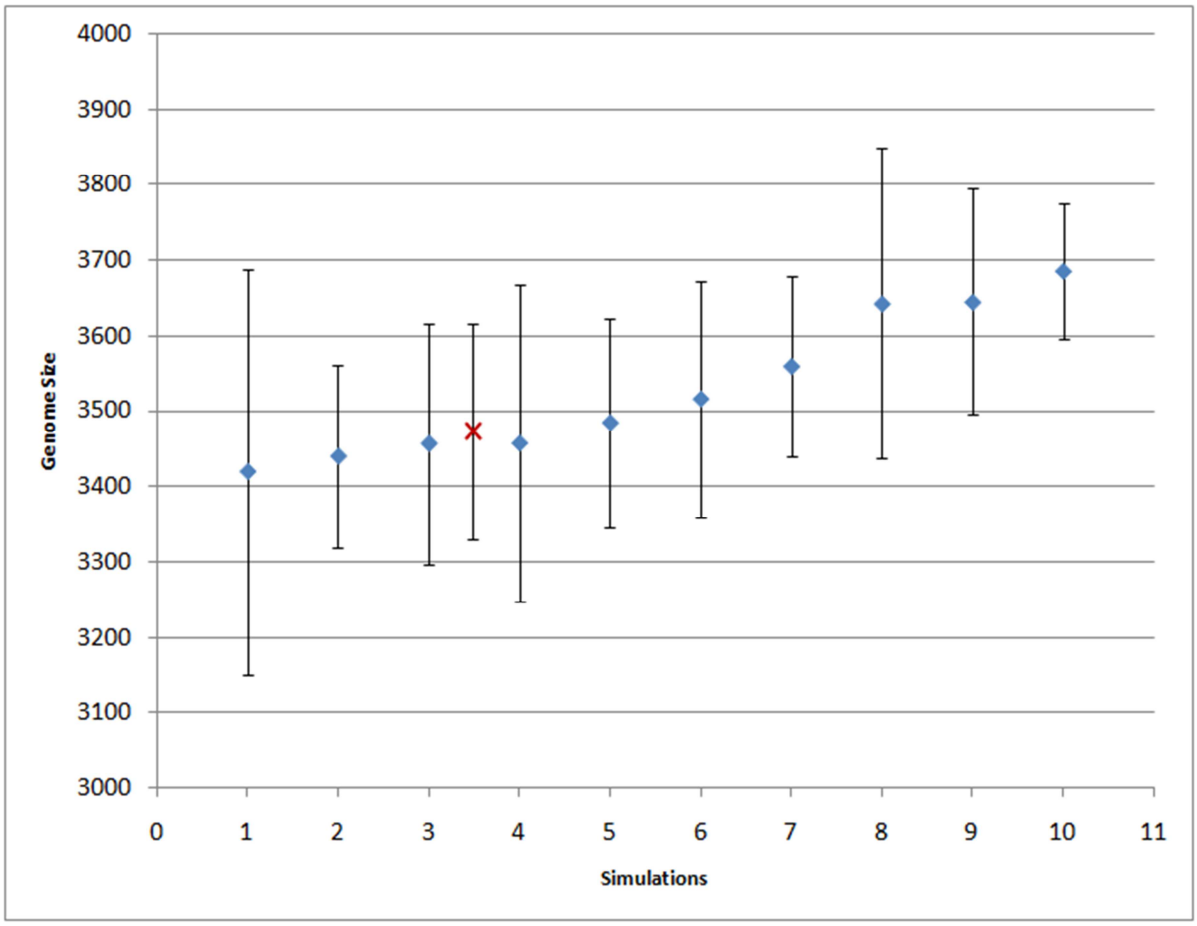

Figure 5.2. Genome size mean and standard deviation of ten simulations and the *Brucella* data set (red cross). **Gene loss, gene insertion and HGT rate are 1e-4, 9e-5, and 1e-4 respectively. The number of genes in LCA is set to 3500 (3000 on the main chromosome, 500 on one plasmid). The simulations have run for 1000 generations.*

The above results are evidence that PEGsim is capable of producing a simulated data set with basic properties that are close to those in a real data set.

Replicon number evaluation

The genome architecture of the simulated genomes can be adjusted by modifying any one or combinations of the following parameters: chromosome split rate, chromosome merge rate, plasmid loss rate, plasmid merge rate, plasmid split rate, plasmid gain rate. As expected, lower plasmid gain and split rates will generate genomes with fewer replicons, and higher plasmid split and gain rates will generate genomes with more replicons. With some parameters properly set, we were able to obtain simulations with the number of replicons similar to the *Rhizobiales* data set. Ten randomly selected such simulations and the Rhizobiales data set were plotted in Figure 5.3. The real data point is shown in red.

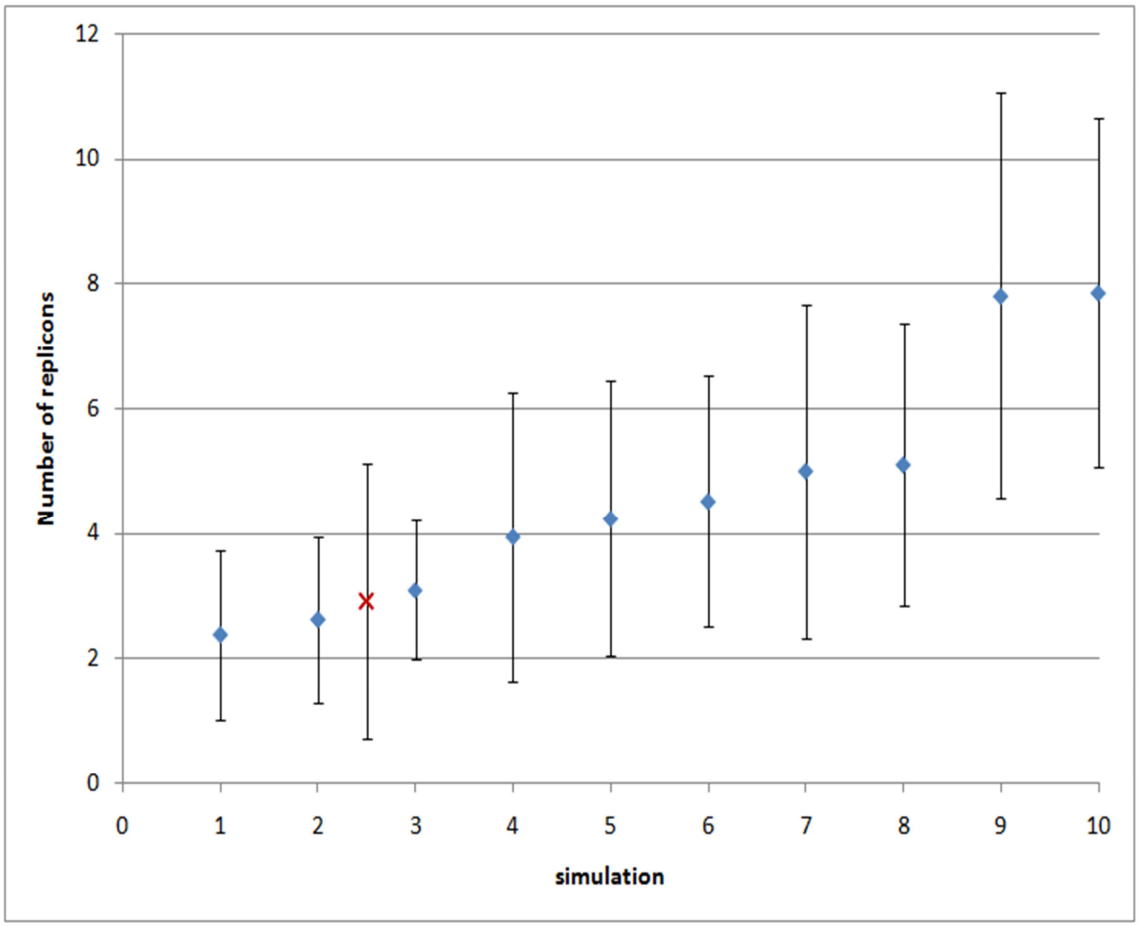

Figure 5.3. The mean and standard deviation of the number of replicons in ten simulations and the *Rhizobiales* data set (red cross). **Chromosome split rate, chromosome merge rate, plasmid loss rate, plasmid merge rate, plasmid split rate and plasmid gain rate are set to 1e-13, 5e-11, 2e-10, 1e-10,1e-9, and 1e-9 respectively.*

Conserved blocks length evaluation

In addition to homologous genes, related species usually show a higher level of genome conservation in conserved blocks. The length of these conserved blocks roughly follows power-law distributions with long tails [47, 48], which are due to the occurrence of a few long conserved blocks. For example, in Figure 5.4 we show the distribution of conserved

blocks at different lengths shared by *Parvibaculum lavamentivorans DS-1* and *Azospirillum B510 uid32551*. Although there are no conserved blocks of lengths between 17 and 27, there is one of length 28.

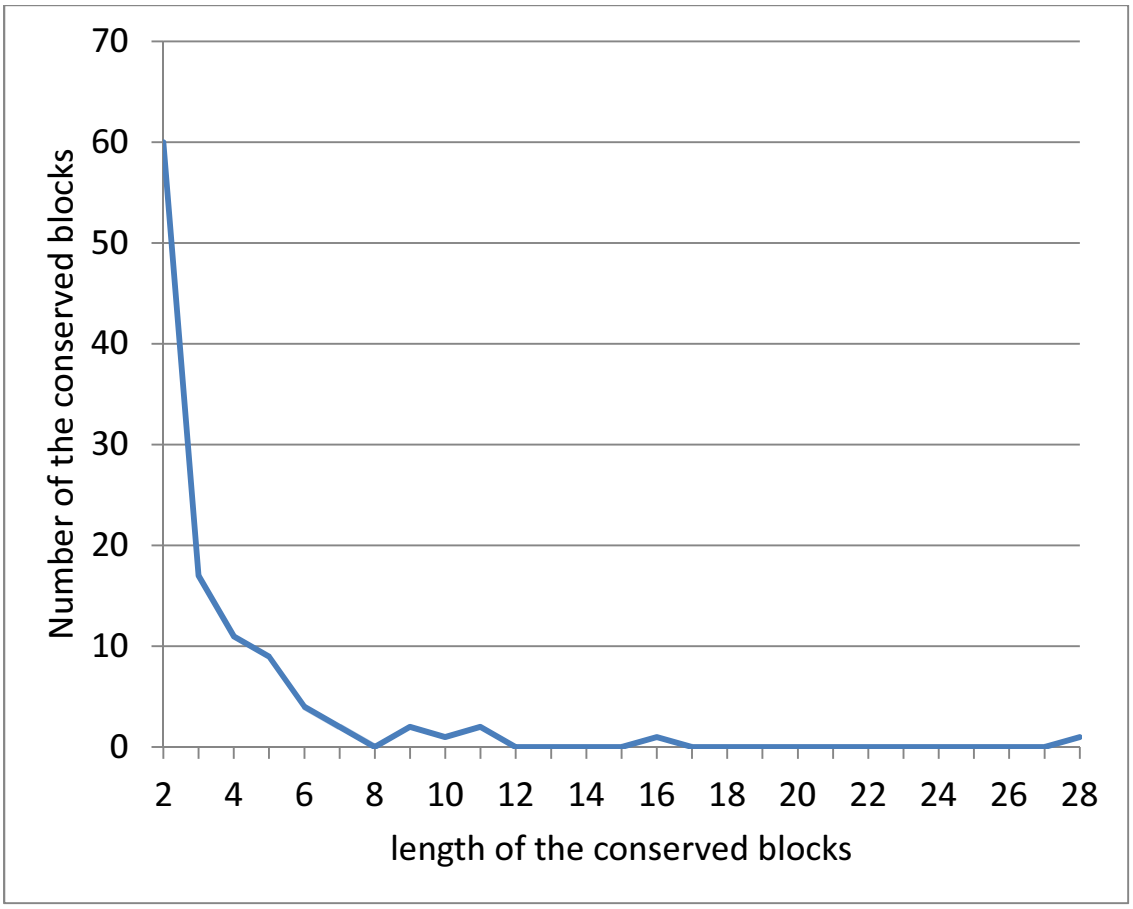

Figure 5.4. Number of conserved blocks shared by *Parvibaculum lavamentivorans DS-1* and *Azospirillum B510 uid32551* at different lengths.

These long tails are difficult to simulate if we just use random events, since the long conserved blocks are of course not random events. We deal with this by creating the concept of *simulated conserved block*, which is implemented by rejecting a certain fraction of evolutionary events that would have disrupted existing blocks of genes. This

allows a few long conserved blocks to “survive” over the course of the simulation, thus replicating what is observed in real genomes. In Figure 5.5 we show the average numbers of conserved blocks at different lengths from ten simulations generated by two different models, one with simulated conserved blocks turned on and one not. All simulations only include two species for simplicity reasons. It is easy to see that the long tail phenomenon is not present in the model without simulated conserved blocks. By customizing the parameters initial conserved gene percentages and initial conserved blocks distribution, different conservation levels can be achieved.

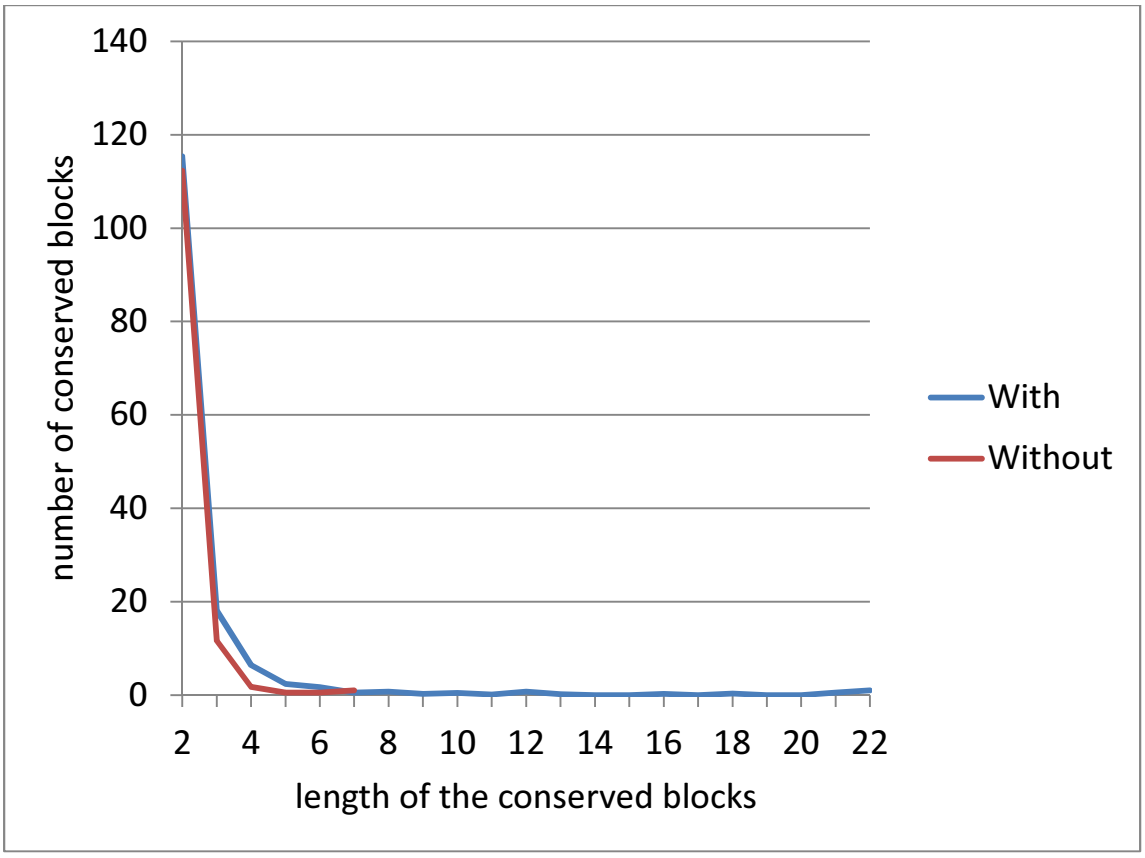

Figure 5.5. Comparison of distribution of the number of syntenic blocks between the model with conserved blocks and the one without.

5.2 Additional Remarks

We have developed the first whole genome simulator for prokaryotes that focuses on gene-scale, replicon-scale, and species-scale events. We have shown that the simulator is capable of producing data with vastly different properties (such as genome size and number of replicons among extant species), mimicking observed properties of real life genomes. By the implementation of conserved blocks, we have managed to overcome the problem that almost no long contiguous gene runs occur in simulated data produced with pure probability models. We have used recently published literature as guidance to set default parameters so that non-expert users can also obtain high quality simulation. PEGsim is also highly customizable for users with the necessary expertise. Together with the simulator code, we also provide various scripts that measure different properties of the simulated data to assist parameter setting by the end users, if they choose to do so. A master script that enables running multiple simulations in parallel is also included.

By using two separate streams of random numbers, the simulator separates events that affect the tree topology from all other events, so the end user can have simulations with the same phylogenetic tree but different gene- and replicon- scale events. A typical use case is for the user to first disable all other events except for speciation events to get a satisfactory (according to some criterion) tree. With all other events disabled, this process is extremely fast. Then, the user can rerun the simulator with the seed that determined the tree topology and all other evolutionary events enabled to produce multiple simulations.

It is also possible to obtain simulations with events at all scales by combining PEGsim with other existing sequence-scale simulators, such as Dawg [29], SIMULATOR [30], or

SIMGRAM [31]. PEGsim can be run first to provide a scaffold of simulated genomes and then specialized sequence-scale simulators can be run according to the property of the region, such as genic or intergenic. A complete simulation can be achieved by merging these two pieces of information.

PEGsim as here described is a first version. As such, it can be improved in a number of ways. We are working on PEGsim in an iterative fashion so we can make sure that the basic structure of the simulator is always stable, and more features can be added in a controlled fashion.

A version of this chapter was published as Yang K, Setubal JC: A Whole Genome Simulator of Prokaryote Genome Evolution. In: *Proceedings of the 2011 ACM Conference on Bioinformatics, Computational Biology and Biomedicine*: 2011 (extended abstract) [28].

Chapter 6

REGEN: Ancestral Genome Reconstruction for Bacteria

In this chapter, we describe the ancestral genome reconstruction system that we developed, called REGEN, and we evaluate it using simulated and real data sets.

6.1 System and Methods

6.1.1 System overview

REGEN has several components. Figure 6.1 shows all the major components and their relationships to each other. The assumed inputs and the outputs were described in Chapter 3.

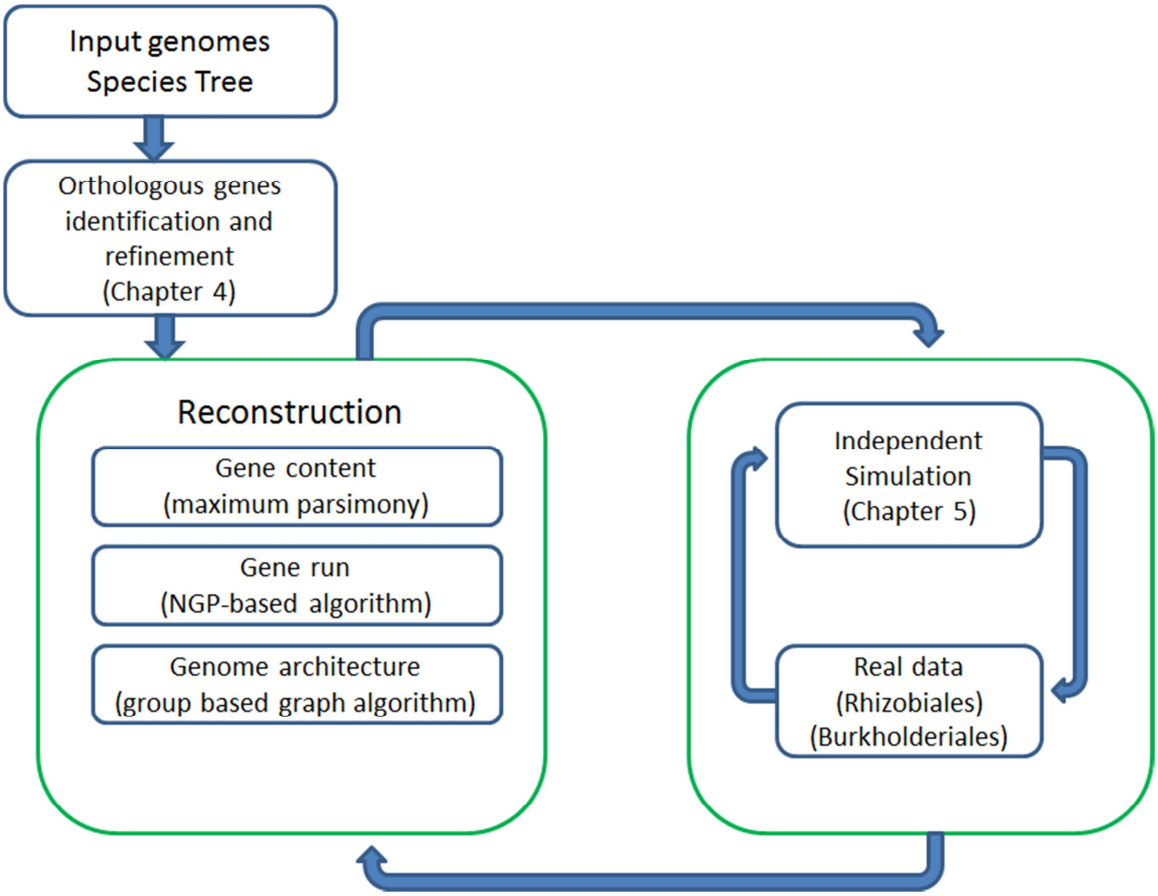

Figure 6.1. Overview of all major components in REGEN

6.1.2 Species tree reconstruction

REGEN needs a reliable species tree as input. In cases where such a tree is not available, a phylogenomic tree based on the multiple sequence alignment of the concatenated sequences from thousands of protein sequences can be built. Here we briefly describe the methodology used to build this tree, which follows the supermatrix approach [49].

An all-against-all BLAST [37] search between all protein sequences annotated in the input genomes is performed. The BLAST output is then fed to OrthoMCL [22] to identify orthologous gene families. Families with at most one member in all the species are

selected for the tree reconstruction. MUSCLE [50] is used to perform multiple sequence alignment for each family, and Gblocks [51] is used for alignment trimming. Trimmed alignments are concatenated and fed to RaxML [52] for final species tree reconstruction evaluation with bootstrap scores.

6.1.3 Homologous gene family identification and refinement

We used OrthoMCL for homologous gene family identification, and the result is further refined by the program described in Chapter 4. When a species tree needs to be built (see previous section) we can use the results of OrthoMCL for both tree construction and homologous gene family refinement.

6.1.4 Genome preprocessing

All genomes are preprocessed so that each replicon is an ordered array of genes, which are represented by the orthologous gene family ID consisting of both the original orthologous gene families identified by OrthoMCL and the ones produced by the refinement module.

6.1.5 Ancestral genome reconstruction

Our ancestral genome reconstruction method is based on the concept of neighboring gene pairs (NGPs), first proposed in [5]. An NGP is a pair of genes that are physically adjacent to each other on a replicon. The key idea is to first identify NGPs in input genomes. Then we infer the occurrence of these NGPs in the ancestral genomes. The basic assumption of

the method is that if adjacent homologous genetic loci are observed in both child species, then it is highly likely that they are also adjacent in the parent species.

The reconstruction is done by a maximum likelihood (ML) method as implemented in BayesTraits [53]. The gene pair occurrence likelihood cutoff that determines what gene pairs are present in an ancestral genome is an important parameter and directly determines the number and length of the reconstructed gene runs. On the other hand, the gene occurrence likelihood cutoff that determines what genes are present in an ancestral genome has less impact on the results, since singleton genes cannot be placed in reconstructed gene runs. A maximum parsimony (MP) based reconstruction is also possible by a slightly modified version of the method described in [5].

In our implementation, each gene is represented by two symbols, one for each end. This notation allows us to encode both the adjacency and orientation information for each gene. This two-node notation also reduces the complexity of the assembly process as described later. Each adjacent gene pair is treated as a feature for a genome, and the status of such features on the ancestral genomes is reconstructed using the same method as described in the gene content reconstruction. After the successful reconstruction of all the NGPs, the following algorithm is designed to reconstruct gene runs for each ancestral genome.

The algorithm starts from a random pair, identifies all other pairs that may be connected through an iterative fashion, and builds an undirected connected graph with all these pairs. Each edge was weighted as the reciprocal of the probability of having the particular NGP in the ML based reconstruction and 1 in the MP based reconstruction. All edges

connecting the ends of a single gene have weight set to 1 in both cases. Then, the algorithm identifies the minimum spanning tree (MST) in this graph by Kruskal's algorithm [54] to obtain a subgraph without cycles. The Bellman-Ford algorithm [55] is then run on the MST to calculate scores for paths between all node pairs. A legitimate path connecting two outer nodes with the lowest score is then identified and recorded as a reconstructed gene run. A path is legitimate if and only if inter-gene edges and intra-gene edges interleave. All nodes included in the path are removed following the identification and the original MST is reduced and may split into two or more fragments. The Bellman-Ford algorithm is run on each fragment and the process repeats recursively until all nodes are removed or a new fragment consists of only one gene.

The establishment of replicon inheritance relationship is based on the following graph-based algorithm, designed to utilize the concept of a group, which is defined as a collection of genes that share the same inheritance pattern. Genes are considered co-inherited if they reside together on a single replicon in both genomes. For example, if genes *a*, *b*, and *c* are on one replicon in both species *X* and *Y*, then they are considered to be in the same group. The reconstruction assumes that co-inherited genes are more likely to be on the same replicon in the ancestral genome because the probability of having multiple genes relocating to the same replicon through independent evolutionary events is low. The idea behind the algorithm is first to divide the genes on the replicons in the extant species into co-inherited groups and then determine which groups are likely to be on the same replicon in the ancestral species and finally merge the groups back into replicons according to the linkages established during the reconstruction process.

Essentially, only genes that share some inheritance pattern in both the out-group and in-group species are merged into a replicon, which turns out to be a quite stringent criterion.

The algorithm is explained with the following example, in which the four extant species are named S1, S2, S3, and S4 and the ancestral species A1, A2 and A3 respectively (see Figure 6.2). For simplicity, all main chromosomes in the four extant species are named C and the plasmids are named P1 and P2. Here, we will decide the genome architecture of A2, using A3 as outgroup. Notice that the genome architecture of A1 cannot be determined without adding more species as outgroups. The algorithm starts by computing co-inherited gene groups for each ancestral species in the tree in a bottom-up fashion. In our example, four groups are identified for A2 and two for A3. Group graphs GA2 and GA3 are created for the two ancestral species, with each group as a vertex. Edges are added if two groups share a replicon in their co-inheritance pattern such as G1(C C) and G2(P1 C) (shared C in S2). Then, the relationship between groups in GA2 and GA3 is computed, and edges are added if the number of shared genes exceeds a certain cutoff, denoted by dark red edges. For each connected component in the outgroup group graph, which is GA3 in the example, we identify all vertices in the target species group graph. For these identified vertices, we will merge the ones that are connected back into replicons. Any unmerged group will form its own replicon, such as R2. Final genome architecture for A2 is shown by green ovals.

After replicon reconstruction, all genes are tagged with their own replicon information.

6.1.6 Reconstructed replicon merge

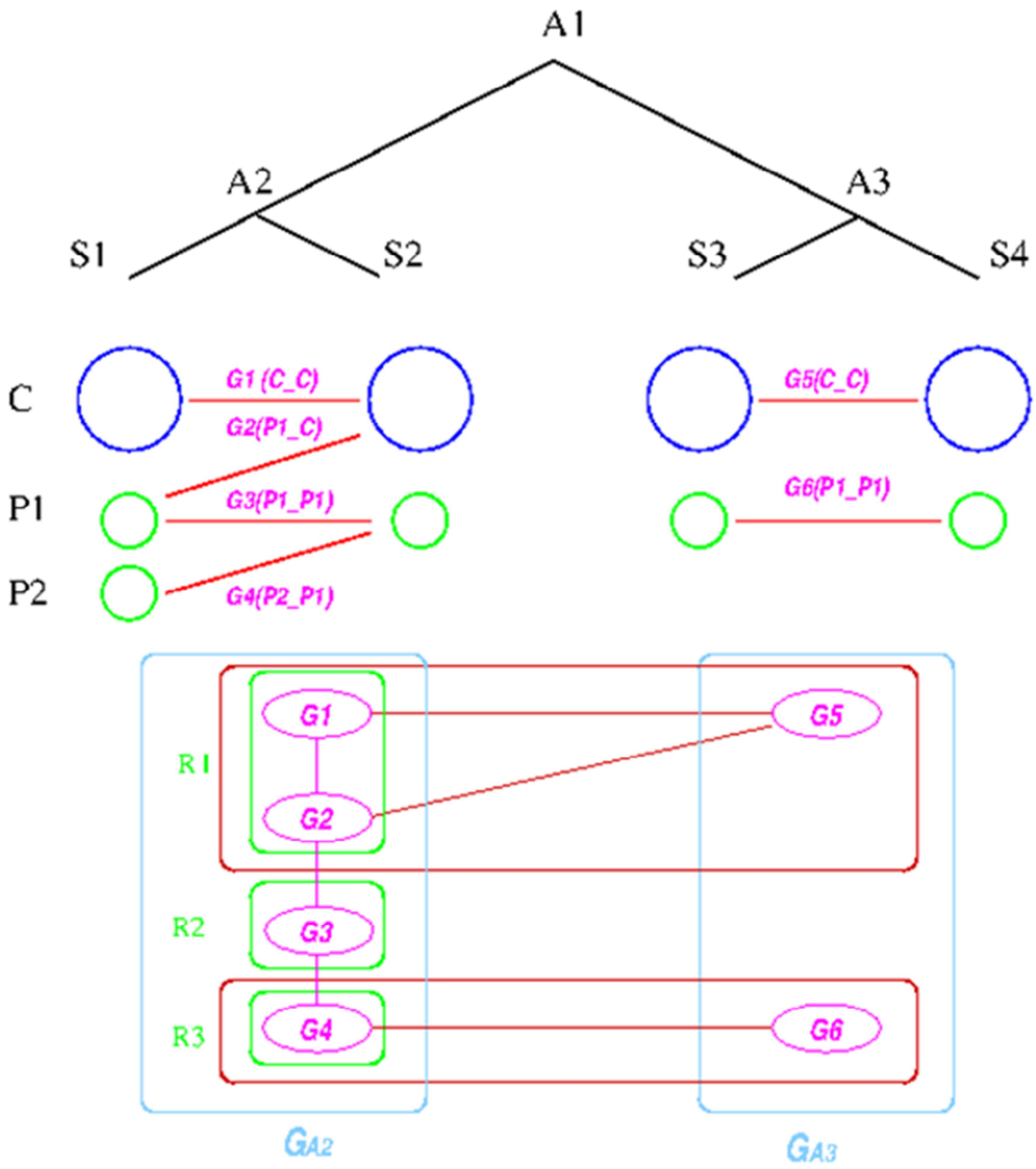

Figure 6.2. Replicon architecture reconstruction example. *Blue circles represent main chromosomes, green circles plasmids, and purple ovals gene groups. Red boxes represent identified connected components in the group graph and green box final replicon architecture reconstruction result.*

During the application of the above algorithm to both real and simulated data, we noticed that it tends to produce more replicons than there really are. We also noticed that many long reconstructed gene runs contain genes that have been assigned to different replicons. Based on these observations, we designed an extra step that merges replicons based on the discrepancy of replicon information for genes in the gene runs.

The algorithm starts by selecting a set of reconstructed gene runs with a length limit, which is set to 4 genes for the data shown in this dissertation. Then it checks for discrepancies of the gene location information in this set of gene runs. Discrepancy is defined as genes on the same gene run that are assigned to different replicons. From each reconstructed gene run in this selected collection, we evaluate the relative signal strength of replicon merging, which is defined as

$$relative\ signal\ strength = \frac{N_b}{N_a} \quad ,$$

where N_a is the number of genes assigned to the most frequently occurring replicon and N_b is the number of genes assigned to another replicon.

Gene runs with extremely low signals are ignored. We then assign the length of the gene run as the strength of the merging proposal supported by this specific gene run. The strength of all gene runs for merging the same pair of replicons are summed and the result is defined as the absolute signal strength of the merging proposal at the species level. All merging proposals are gathered together for all ancestral species, and a K-means clustering is performed on both absolute and relative signal strength, with $K = 2$. The values that divide the result clusters are chosen as the line between accepting or rejecting

merging proposals. The algorithm proceeds in a bottom-up fashion and ends when all merging proposals are either accepted or rejected.

6.1.7 Chromosome restoration

With the replicons for each ancestral genome being reconstructed, it is time to distinguish chromosomes from plasmids. The notion of chromid [56] is not considered here. This process is carried out using core genes. The main chromosome is assigned to the replicon with the most core genes.

For secondary chromosome assignment, a minimum number of core genes (5% of the total number of core genes by default) have to reside on the replicon.

6.1.8 Ancestral evolutionary event reconstruction

By comparing the gene runs and gene content between parent and child species, we can infer a large number of different evolutionary events on both the gene and replicon scales, such as gene loss, gene gain, replicon merge, and replicon loss. We can even infer gene reversal events, if they happened within a reconstructed gene run.

6.1.9 Ancestral gene run and genome functional annotation

Kyoto Encyclopedia of Genes and Genomes (KEGG) [57] was used as the source of functional annotation. To determine the potential phenotypic features of an ancestral species, we need to first determine the function of as many of the genes that it possesses as possible. To achieve this, we assign the most frequently occurred functional annotation among all family members to the function annotation for the orthologous gene family.

Multiple functions are assigned when there is a tie. The determined function is later transferred to the gene in the ancestral genome. After the completion of annotating as many genes in the ancestral genome as possible, we determine possible ancestral phenotypic features by examining the gene content with its functional annotations in the ancestral genome.

Due to the close resemblance between reconstructed consecutive gene runs and operons in bacterial genomes, not only did we use the annotated genes to infer the functional roles played by some gene runs, but we also validate these reconstructed gene runs by checking the consistency among the members they contain.

6.1.10 Genomes

The group of Rhizobiales species was chosen not only because of their complex genome architecture, as shown in Table 6.1, but also because of the fact that they contain secondary chromosomes, which is not common among bacteria. The 22 species from the Rhizobiales order include *Agrobacterium tumefaciens C58 Cereon, Agrobacterium vitis S4, Agrobacterium radiobacter K84, Azorhizobium caulinodans ORS 571, Bartonella henselae Houston- 1, Beijerinckia indica ATCC 9039, Bradyrhizobium japonicum, Brucella suis 1330, Mesorhizobium BNC1, Hyphomicrobium denitrificans ATCC 51888 uid33261, Methylobacterium chloromethanicum CM4, Methylocella silvestris BL2, Nitrobacter hamburgensis X14, Ochrobactrum anthropi ATCC 49188, Oligotropha carboxidovorans OM5, Parvibaculum lavamentivorans DS-1, Rhizobium etli CFN 42, Rhodomicrobium vannielii ATCC 17100 uid38253, Rhodopseudomonas palustris BisA53, Sinorhizobium meliloti, Starkeya novella DSM 506 uid37659,* and *Xanthobacter*

autotrophicus Py2. *Azospirillum B510 uid32551* is chosen as an outgroup. The choice of the outgroup species was made based on the phylogenetic tree presented in [58]. All genome sequences were downloaded from the NCBI Genbank FTP site.

Table 6.1. Genome architecture for Rhizobiales and integer ID assigned to each genome

Species Name	Integer ID	# of chromosomes	# of plasmids
Agrobacterium_tumefaciens_C58_Cereon	1	2	2
Agrobacterium_radiobacter_K84	2	2	3
Agrobacterium_vitis_S4	3	2	5
Azorhizobium_caulinodans_ORS_571	4	1	0
Azospirillum_B510_uid32551	5	1	6
Bartonella_henselae_Houston-1	6	1	0
Beijerinckia_indica_ATCC_9039	7	1	2
Bradyrhizobium_japonicum	8	1	0
Brucella_suis_1330	9	2	0
Mesorhizobium_BNC1	10	1	3
Hyphomicrobium_denitrificans_ATCC_51888 _uid33261	11	1	0
Methylobacterium_chloromethanicum_CM4	12	1	2
Methylocella_silvestris_BL2	13	1	0
Nitrobacter_hamburgensis_X14	14	1	3
Ochrobactrum_anthropi_ATCC_49188	15	2	4
Oligotropha_carboxidovorans_OM5	16	1	0
Parvibaculum_lavamentivorans_DS-1	17	1	0
Rhizobium_etli_CFN_42	18	1	6
Rhodomicrobium_vannielii_ATCC_17100_uid 38253	19	1	0
Rhodopseudomonas_palustris_BisA53	20	1	0
Sinorhizobium_meliloti	21	1	2
Starkeya_novella_DSM_506_uid37659	22	1	0
Xanthobacter_autotrophicus_Py2	23	1	1

6.2 Results and Discussion

6.2.1 Results based on Simulation

With the simulator described in Chapter 5, we are able to compare results with different settings and make an informed choice on the parameter settings for the real data set.

Due to the amount of time required to generate simulated data and to perform reconstructions, we set the LCA with a small genome consisting of a main chromosome of 1000 genes and a plasmid of 200 genes. Twenty simulations were conducted with the same phylogeny, which contains 19 extant species of interest and 2 out-groups.

We compared reconstruction produced by the MP and ML methods with gene pair cutoff set to 0.75, 0.8, 0.85, 0.9, 0.95, and 0.97. Singleton gene occurrence cutoff is set to 0.9 in all ML reconstructions.

All the numbers shown below are averages calculated from all of the same-setting reconstructions of the 20 simulated data sets. Evaluation with simulated data includes genome coverage, longest reconstructed gene run length, conserved block reconstruction, gene pair precision versus recall measure, and replicon reconstruction accuracy. Based on all the benchmarks we obtained using simulated data, we set the gene pair cutoff to 0.9 for the system.

Genome Coverage

By comparing the reconstructed gene runs of the LCA to the true genome, we are able to calculate how much of the genome is covered by the reconstructed gene runs. The result is shown in Figure 6.3.

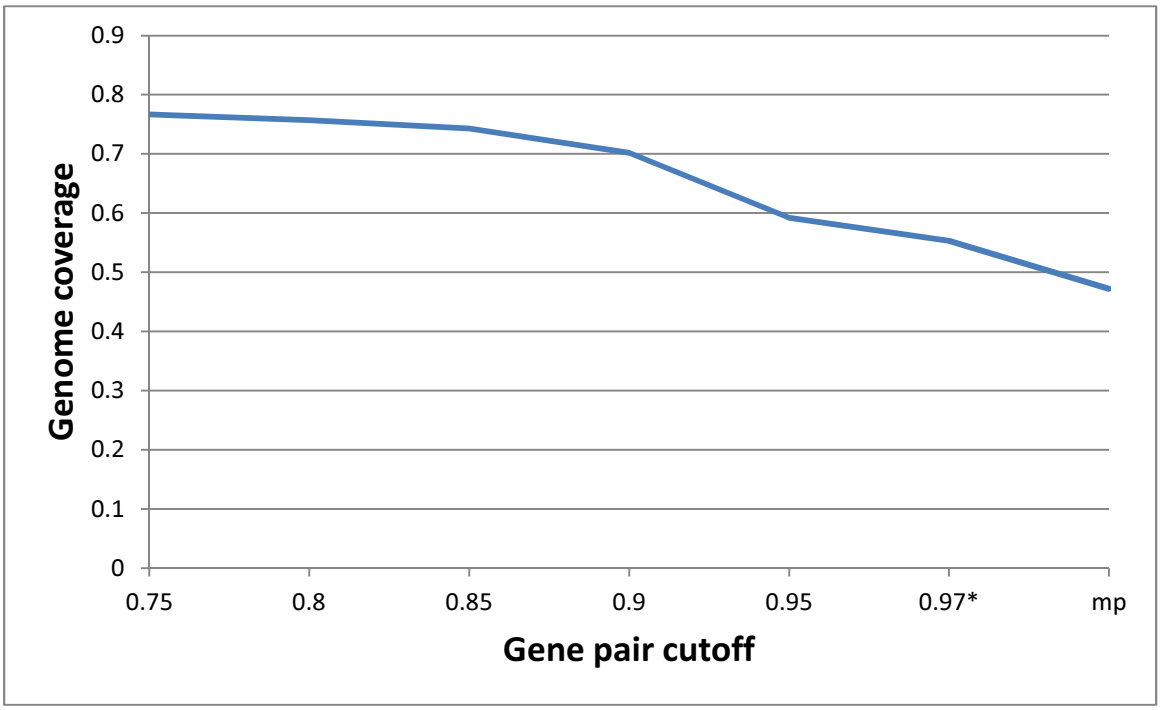

Figure 6.3. Genome coverage achieved by reconstructions at different gene pair cutoff.

Setting the gene pair occurrence cutoff to a lower value naturally results in more gene pairs, which then cover more of the genome. It is worth noticing that the significant coverage decrease is not observed until the setting reaches 0.95 and MP achieves the least genome coverage.

Longest reconstructed gene run length

We then looked into the length of the longest reconstructed gene runs. With a similarity to genome assembly problem, the longest gene run is of particular interest. Figure 6.4 shows the length of the longest reconstructed gene run at different settings. We also show the length of the longest subrun that can be entirely mapped to the reference genome as a comparison.

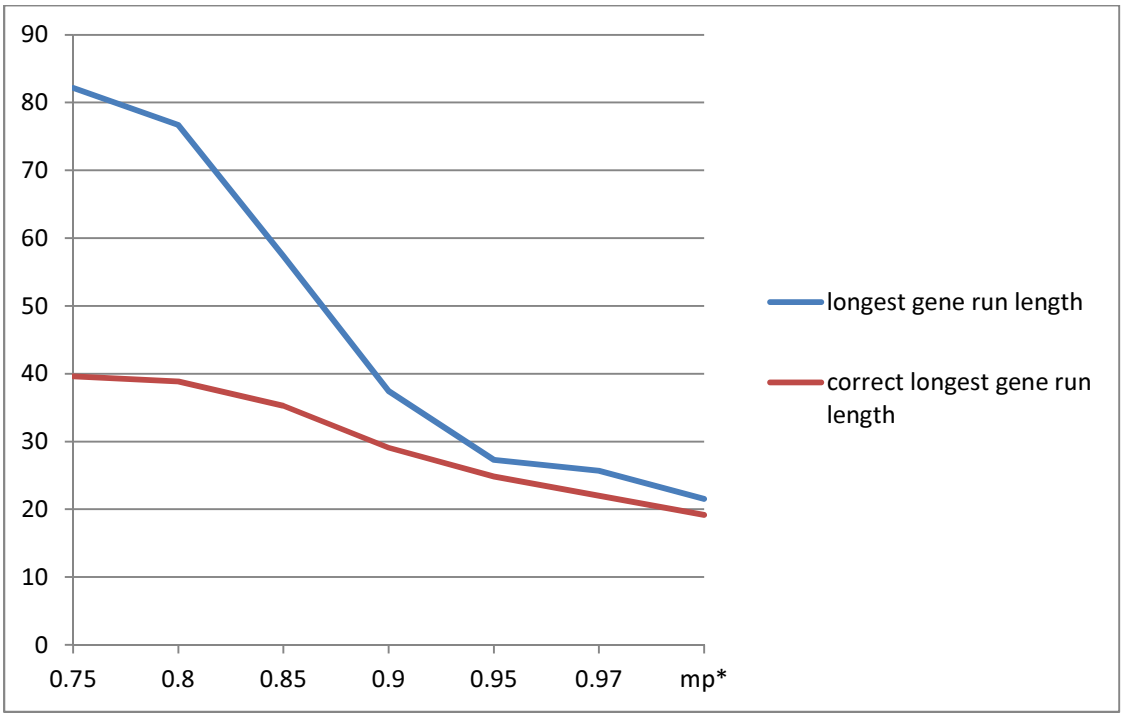

Figure 6.4. Longest gene run length and correct longest gene run length in the reconstructions at different cutoff.

Conserved Blocks Reconstruction

Conserved blocks are conserved gene runs of the genome that carry important functions and thus are more conserved than other parts of the genome. We are interested in determining how many of the conserved blocks can be recovered by the reconstructed gene runs in the ancestral genome. Although conserved blocks and gene runs are different concepts, it is reasonable to assume that if a conserved block exists in most if not all the extant species, there should be a gene run with the conserved block in the genome of the LCA of this group.

Figure 6.5 shows the comparison of the percentages of conserved blocks that have been completely reconstructed or missed in different reconstructions as well as the distribution on the percentage of conserved blocks that might have been partially reconstructed.

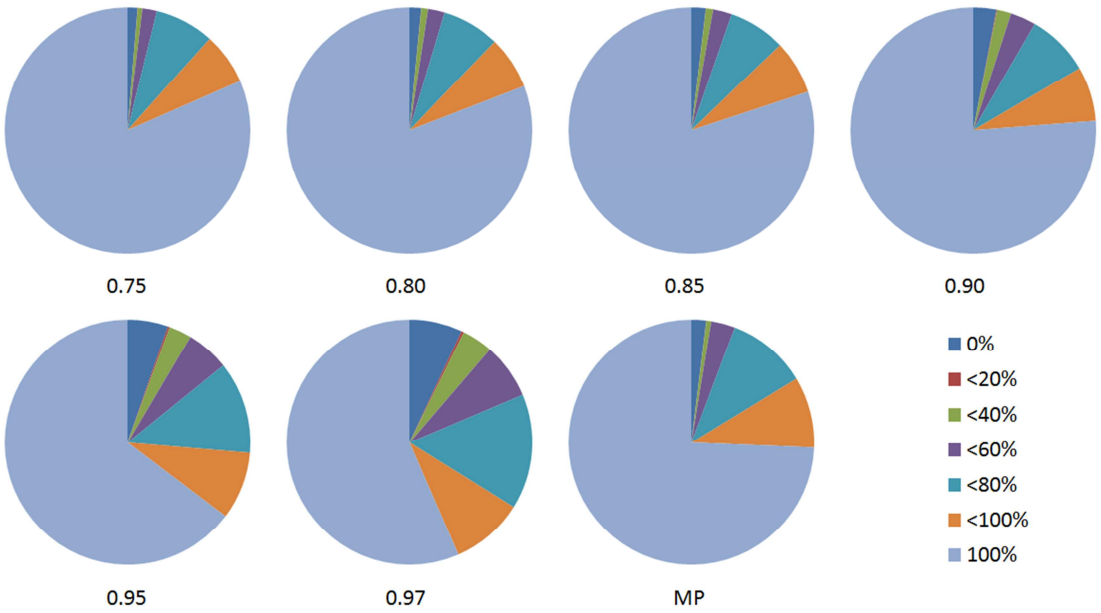

Figure 6.5 partially reconstructed conserved blocks percentage distribution. *0% means complete absent in the reconstruction. <x%: less than x% of the conserved block*

(measured in number of genes) was reconstruction. 100%: the conserved block is entirely reconstructed.

One striking observation here is that even though MP-based reconstructions have shown lower performance in other categories, such as longest gene run length and genome coverage, it appears to be able to reconstruct conserved blocks fairly well. We hypothesis that the reason is conserved blocks underwent fewer evolutionary events. Other studies also suggested that MP based reconstruction/phylogenetic tree construction performs well with closely relate species and ML based methods usually perform better with more distant species [59].

Gene Pair Precision and Recall

The final gene-scale assessment we performed on the reconstructions is the gene pair reconstruction precision and recall test. We compared all the reconstructed gene pairs for each ancestral genome to the actual genomes generated in the simulation and calculated the precision and recall as:

$$\text{precision} = \frac{tp}{tp + fp}$$

$$\text{recall} = \frac{tp}{tp + fn}$$

where *tp* represents the true positive count, *fp* represents the false positive count, and *fn* represents the false negative counts. The results are plotted in Figure 6.6. The figure shows that setting the gene pair cutoff too low results in low precision, while setting it

too high severely affects recall. A good balance of precision and recall is achieved for gene pair cutoff between 0.85 and 0.9. Since reconstruction confidence is an important factor to judge a given reconstruction, 0.9 is selected as the gene pair cutoff for the rest of the study.

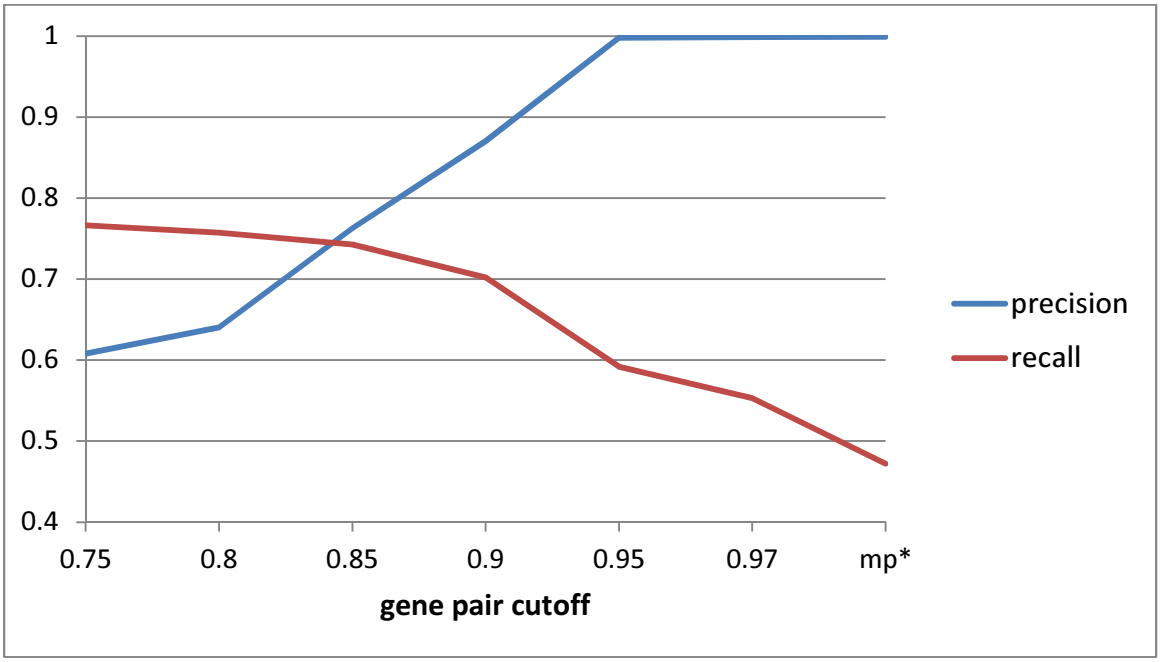

Figure 6.6. Precision and recall for different reconstructions.

Replicon reconstruction accuracy

As the first ancestral genome reconstruction system including replicon scale reconstruction, determining the accuracy of such reconstructions is important. With the simulated data, we are able to accurately measure the performance of the system with the following metrics. For an ancestral genome, we defined a replicon *matched* if there is a reconstructed replicon that shares a large number (60%) of genes with it, otherwise *missed*. For a reconstructed ancestral genome, we defined a replicon *extra* if it cannot be

mapped to any replicon in the corresponding ancestral genome or *partial* if it is mapped to an already matched replicon.

The four measures are plotted in Figure 6.7. Gene pair cutoff and gene cutoff were set to 0.9 with the consideration of all the information retrieved from simulation tests above.

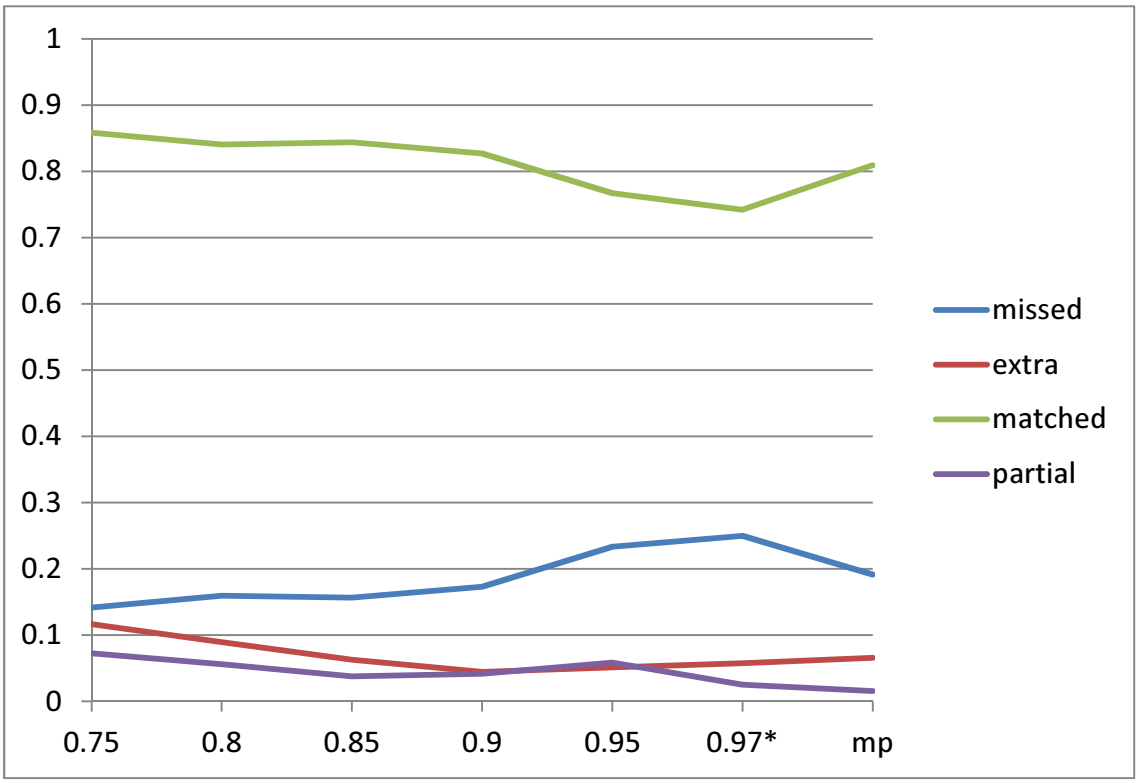

Figure 6.7. Fraction of different scenarios for replicon reconstruction evaluation for different reconstructions.

Figure 6.7 clearly shows that the fraction of matched replicons starts to drop when the gene pair cutoff approaches 0.95, with a corresponding increase in the fraction of missing replicons (the fractions of missed and matched replicons sum to 1). The other two

measures showed inconsistency but are relatively stable across all different settings. Based on all the results generated from simulated data, we believe 0.9 is the best cutoff to adopt.

6.2.3 Results on real genome data

Phylogenomic tree

The phylogenomic tree reconstructed with 109273 genes (4751 orthologous gene families in 23 genomes) for the *Rhizobiales* data set is shown in Figure 6.8. Phylogenomic trees are usually considered more reliable than phylogenetic trees, which are usually constructed using one gene or a very small number of genes, but they can be difficult to build due to the large amount of data involved and high computational cost. Each extant species is assigned an integer ID, which is the appended integer in the species name in the tree, so we can assign an easy and self-explanatory ID for each ancestor species. The ancestor ID reflects both child species. For example, ancestor 14 8 20 is the LCA of species 14, 8, and 20. The complete species to integer ID mapping is given in Table 6.1.

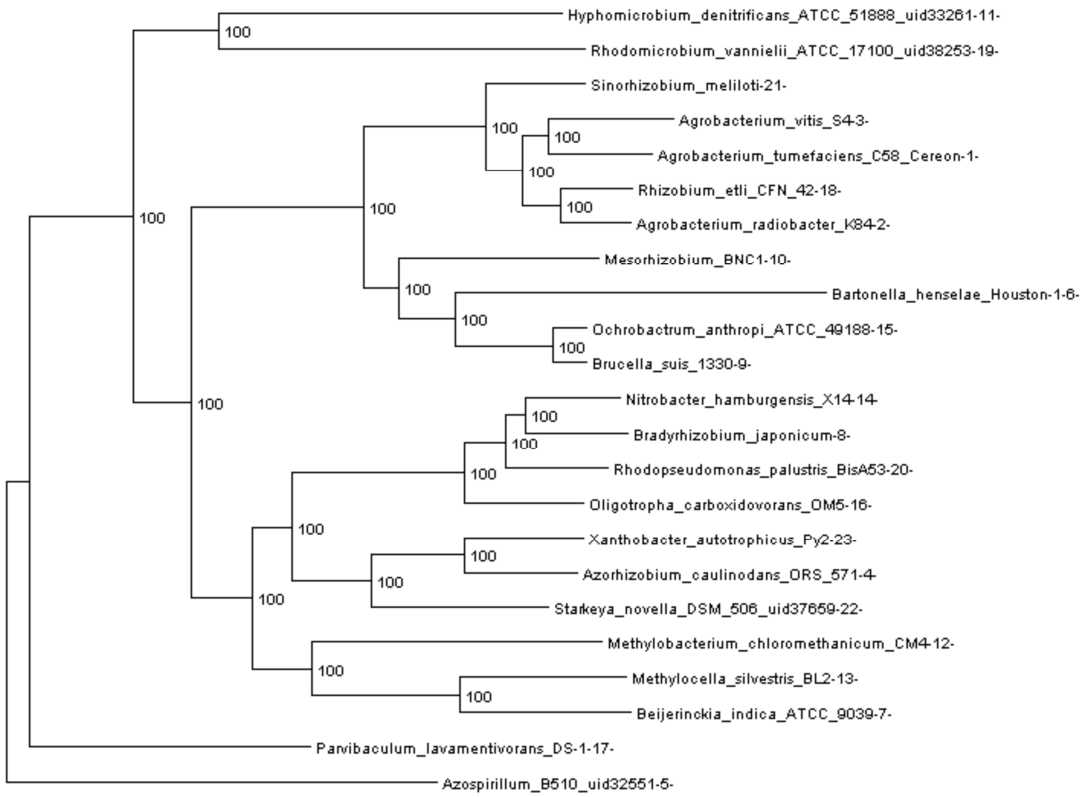

Figure 6.8. Phylogenomic species tree for the *Rhizobiales* dataset. *The number shown at each branching point is the bootstrap score computed by RAxML (100 runs). In this case, all numbers are 100, suggesting that the tree is robust.*

Orthologous gene identification and refinement

OrthoMCL identified 8,563 orthologous gene families, including 53,677 genes that could be used directly for reconstruction. Gene families that are present in only one species were omitted. OrthoMCL also reported 3,125 mixed gene families defined as homologous gene families containing paralogous genes, totaling 38,396 genes. These families underwent a refinement process (Chapter 4) and, at a conservative p-value of 0.01, 3,892 orthologous gene families containing 18,606 genes are obtained. In total,

12,455 orthologous gene families with 72,283 genes were used as input for the reconstruction process.

Ancestral gene content reconstruction

To be consistent with the gene-pair reconstruction cutoff, all genes tagged with < 0.9 probability are removed from further analysis. Details of the gene content reconstruction for each ancestor can be found in Table 6.2.

Table 6.2. Gene content reconstruction

Ancestor ID	Gene on chromosomes	Genes on plasmids	total
11_19_21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7_17	1435	219	1654
11_19_21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7	1446	569	2015
21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7	1457	760	2217
14_8_20_16_23_4_22_12_13_7	1272	988	2260
21_3_1_18_2_10_6_15_9	1955	863	2818
14_8_20_16_23_4_22	1287	1082	2369
21_3_1_18_2	2549	1627	4176
3_1_18_2	2464	1888	4352
14_8_20_16	2560	480	3040
10_6_15_9	1754	257	2011
12_13_7	1245	557	1802
23_4_22	2603	211	2814
6_15_9	2146	98	2244
14_8_20	2940	431	3371
14_8	2479	390	2869
3_1	3507	660	4167
18_2	4941	642	5583
13_7	1636	247	1883
23_4	2271	263	2534
15_9	3358	462	3820
11_19	1221	136	1357

Ancestral gene run reconstruction

The ancestral genome reconstructions are achieved through the use of gene runs and singleton genes. Compared to singleton genes, gene runs provide information on the order of a certain number of genes.

Local synteny or conserved block information is extremely useful in genomics studies, due to their correspondence to operons or modulons. Table 6.3 lists the status of the reconstructed gene runs in the *Rhizobiales* data set. The last two columns show the absolute number of genes in gene runs and the respective percentage. It is easy to see that the quality of the reconstruction for the ancestor improves with the similarity between the genomes of child species. Higher similarity results in longer gene runs, which cover more genes, leaving fewer genes to be singleton genes in the genome. For example, the ancestral species 15_9 has its longest gene run with 121 genes and about 95% of its genes are in gene runs. On the other hand, the longest gene run in the ancestral species 13_7 only reaches 28 genes and about 33% of all its genes are singleton genes.

Table 6.3. Contiguous gene run reconstruction overview of the *Rhizobiales* group. Length of the gene runs is measured in genes. The number-of-genes column shows the total number of genes on all gene runs and the percentage column shows the coverage of the gene runs.

Ancestor	# of gene runs	longest gene run	# of gene of gene runs	percentage of genes of gene runs
11_19_21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7_17	305	32	1321	79.87%
11_19_21_3_1_18_2_10_6_15_9_14	409	33	1716	85.16%

_8_20_16_23_4_22_12_13_7				
21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7	457	33	1894	85.43%
14_8_20_16_23_4_22_12_13_7	461	33	1869	82.70%
21_3_1_18_2_10_6_15_9	510	49	2394	84.95%
14_8_20_16_23_4_22	497	35	2019	85.23%
21_3_1_18_2	685	44	3688	88.31%
3_1_18_2	681	47	3918	90.03%
14_8_20_16	509	47	2775	91.28%
10_6_15_9	352	33	1561	77.62%
12_13_7	333	17	1024	56.83%
23_4_22	556	35	2336	83.01%
6_15_9	402	36	2153	95.94%
14_8_20	525	31	2513	74.55%
14_8	339	40	2638	91.95%
3_1	483	75	3863	92.70%
18_2	589	136	5280	94.57%
13_7	384	28	1272	67.55%
23_4	502	16	1933	76.28%
15_9	353	121	3615	94.63%
11_19	260	31	821	60.50%

Functional annotation of gene runs

Functional annotation of one particular gene run in the root of the *Rhizobiales* is listed in Table 6.4 as an example, and all other annotations can be found in the supplemental material, along with functional annotation for singleton genes.

Table 6.4. Functional annotation of a particular reconstructed contiguous gene run in the LCA of the *Rhizobiales* group. Consensus column shows the number of genes that have been assigned with the corresponding annotation as well as the total number of genes in the family.

Gene family ID	KEGG Entry	Function class	Definition	consensus
1719	K02387	Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	flagellar basal-body rod protein FlgB	17/17
9901747	K02388	Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	flagellar basal-body rod protein FlgC	17/17
9901380	K02408	Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	flagellar hook-basal body complex protein FliE	17/17
9901964	K02392	Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	flagellar basal-body rod protein FlgG	17/17
1718	K02386	Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	flagella basal body P-ring formation protein FlgA	16/17
9903288	K02394	Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	flagellar P-ring protein precursor FlgI	16/17
1717	not annotated	N/A	N/A	N/A
9904536	K02393	Cellular Processes; Cell Motility;	flagellar L-	16/17

		Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	ring protein precursor FlgH	
1828	K02415	Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035]	flagellar FliL protein	16/16
9904106	K02419	Environmental Information Processing; Membrane Transport; Secretion,system [BR:ko02044],Cellular Processes; Cell Motility; Bacterial motility proteins,[BR:ko02035],Cellular Processes; Cell Motility; Flagellar assembly [PATH:ko02040]	flagellar biosynthetic protein FliP	17/17

Evolutionary history of ancestral gene runs

With the completion of the reconstruction of all ancestral gene runs, it is possible to infer a hypothesis on what has happened to each gene run during a specific evolutionary path by analyzing the shared genes in the gene runs in both parent and child species. One example is shown in Figure 6.9.

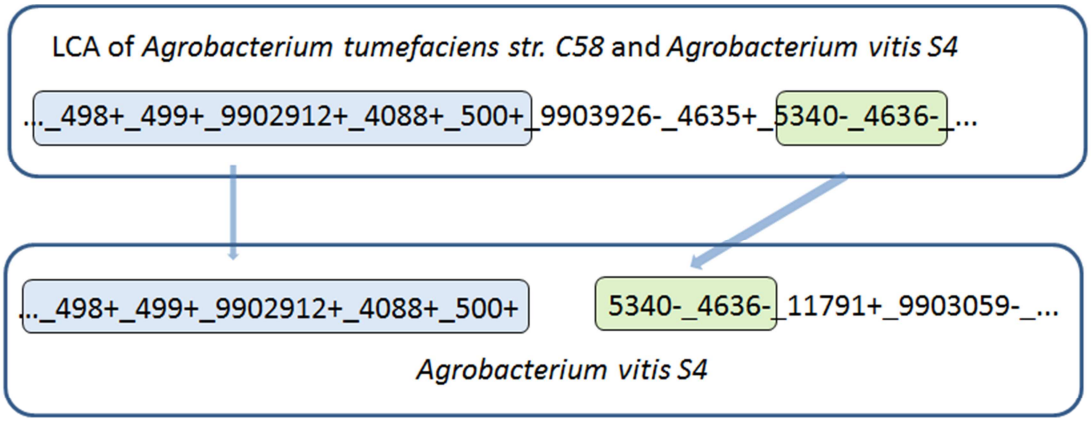

Figure 6.9. A long gene run on the main chromosome split into two smaller fragments during the evolutionary path from the LCA of *Agrobacterium Vitis S4* and *Agrobacterium Tumefaciens C58* to *Agrobacterium vitis S4*. *Each number represents a gene and the underscore represents adjacency. +/- symbols represent the gene orientation determined during the reconstruction. Some genes on both ends are omitted for simplicity.*

All reconstructed scenarios for all evolutionary paths in the tree can be found in the supplemental material.

Replicon reconstruction

Replicon reconstruction is the centerpiece of this study. We reconstructed the genome architecture of all ancestral species through analysis of the gene content of the child species and the outgroup. Based on the reconstructed replicons, replicon-scale evolutionary events can be predicted based on comparison of the genomes along each branch in the tree.

Only two ancestral genomes contain replicons qualified to be secondary chromosomes.

These two ancestors are 15-9 (the ancestor of *Brucella suis* and *Ochrobactrum anthropi*), and 6-15-9 (the ancestor of 15-9 and *Bartonella henselae*). In the path from 10-6-15-9 to 6-15-9, a chromosomal split event divided the main chromosome into two chromosomes and the new secondary chromosome carries a number of core genes. This property may have ensured the survival of this secondary chromosome to the extant species. The distribution of the core genes in all ancestral genomes and secondary chromosome assignment and the distribution of core genes in the extant Rhizobiales species genomes can be found in Table 6.5.

Table 6.5. The distribution of the core genes in all ancestral genomes and secondary chromosome assignment

ancestor	**replicon**	**No. of CG**
6_15_9		
	c1	524
	c2	51
	U	0
10_6_15_9		
	c1	579
	L2	0
	U	0
21_3_1_18_2		
	c1	575
	L3	0
	L5	0
	L6	0
	L7	0
	L9	0
	L10	0
	L11	0
	L12	3
	L14	0
	L16	0
	L18	0

	U	0
21_3_1_18_2_10_6_15_9		
	c1	574
	L3	0
	L4	0
	L5	0
	L7	0
	L8	0
	L10	0
	U	2
14_8_20_16_23_4_22_12_13_7		
	c1	546
	L5	0
	L6	0
	U	0
14_8_20_16		
	c1	577
	L3	0
	L4	0
	L5	0
	U	0
13_7		
	c1	577
	U	0
23_4		
	c1	585
	L4	0
15_9		
	c1	532
	c2	55
	L4	0
	L5	0
	L6	0
	U	0
11_19		
	c1	420
12_13_7		
	c1	501
	R2	0
	L4	0
	U	0

3_1		
	c1	580
	R1	0
	L6	1
	U	0
3_1_18_2		
	c1	580
	R1	0
	R2	0
	R3	0
	R4	0
	L8	0
	L9	0
	U	0
	L7	0
23_4_22		
	c1	558
	U	0
14_8_20		
	c1	560
	R2	0
	L3	0
	U	0
14_8		
	c1	584
	L2	0
	U	0
21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7		
	c1	557
	R1	0
	L5	0
	L7	0
	U	0
18_2		
	c1	584
	R4	0
	L6	0
	L7	0
	U	0
14_8_20_16_23_4_22		

	c1	542
	R1	0
	R2	4
	R3	1
	L5	0
	L6	0
	U	0
11_19_21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7		
	c1	551
	R4	4
	U	0
11_19_21_3_1_18_2_10_6_15_9_14_8_20_16_23_4_22_12_13_7_17		
	c1	545
	U	0

It is also worth noticing that given the definitions adopted in this study, the second largest replicons of *Agrobacterium radiobacter K84* and *Agrobacterium vitis S4* do not qualify as a secondary chromosome, because they do not contain enough core genes, as shown in Table 6.6.

Table 6.6. The distribution of core genes in the *Rhizobiales* data set

Sinorhizobium_meliloti		
	c1	584
	pSymA	0
	pSymB	3
Azospirillum_B510_uid32551		
	c1	527
	pAB510a	18
	pAB510b	0
	pAB510c	16
	pAB510d	17
	pAB510e	9
	pAB510f	0

Rhodopseudomonas_palustris_BisA53		
	c1	587
Beijerinckia_indica_ATCC_9039		
	c1	587
	pBIND01	0
	pBIND02	0
Azorhizobium_caulinodans_ORS_571		
	c1	587
Oligotropha_carboxidovorans_OM5		
	c1	587
Parvibaculum_lavamentivorans_DS-1		
	c1	587
Bartonella_henselae_Houston-1		
	c1	587
Xanthobacter_autotrophicus_Py2		
	pXAUT01	0
	c1	587
Methylocella_silvestris_BL2		
	c1	587
Rhizobium_etli_CFN_42		
	c1	585
	p42a	0
	p42b	0
	p42c	0
	p42d	1
	p42e	0
	p42f	1
Bradyrhizobium_japonicum		
	c1	587
Ochrobactrum_anthropi_ATCC_49188		
	c1	549
	c2	38
	pOANT01	0
	pOANT02	0
	pOANT03	0
	pOANT04	0
Starkeya_novella_DSM_506_uid37659		
	c1	587
Methylobacterium_chloromethanicum_CM4		
	c1	587
	pMCHL01	0

	pMCHL02	0
Nitrobacter_hamburgensis_X14		
	c1	587
	p1	0
	p2	0
	p3	0
Agrobacterium_tumefaciens_C58_Cereon		
	c1	523
	c2	64
	At	0
	Ti	0
Brucella_suis_1330		
	c1	533
	c2	54
Hyphomicrobium_denitrificans_ATCC_51888_uid33261		
	c1	587
Rhodomicrobium_vannielii_ATCC_17100_uid38253		
	c1	587
Agrobacterium_radiobacter_K84		
	c1	587
	c2	0
	pAgK84	0
	pAtK84b	0
	pAtK84c	0
Mesorhizobium_BNC1		
	c1	586
	p1	1
	p2	0
	p3	0
Agrobacterium_vitis_S4		
	c1	580
	c2	7
	pAtS4a	0
	pAtS4e	0
	pAtS4c	0
	pTiS4	0
	pAtS4b	0

Genome architecture evolution reconstruction

The overview of the reconstruction of *Rhizobiales* species with the complete reconstruction process described above is summarized in Figure 6.10, which was automatically generated by REGEN using the dot language in the Graphviz package [60]. It shows that this group of *Rhizobiales* species constantly underwent plasmid split and plasmid merge, which could be true for most bacterial genomes due to the high frequency of recombination. A chromosome can easily pick up genes from a plasmid, which could be a result of a previous lateral gene transfer event. However, it is uncommon for a chromosome to undergo a replicon split and have some core genes migrate away from the main chromosome.

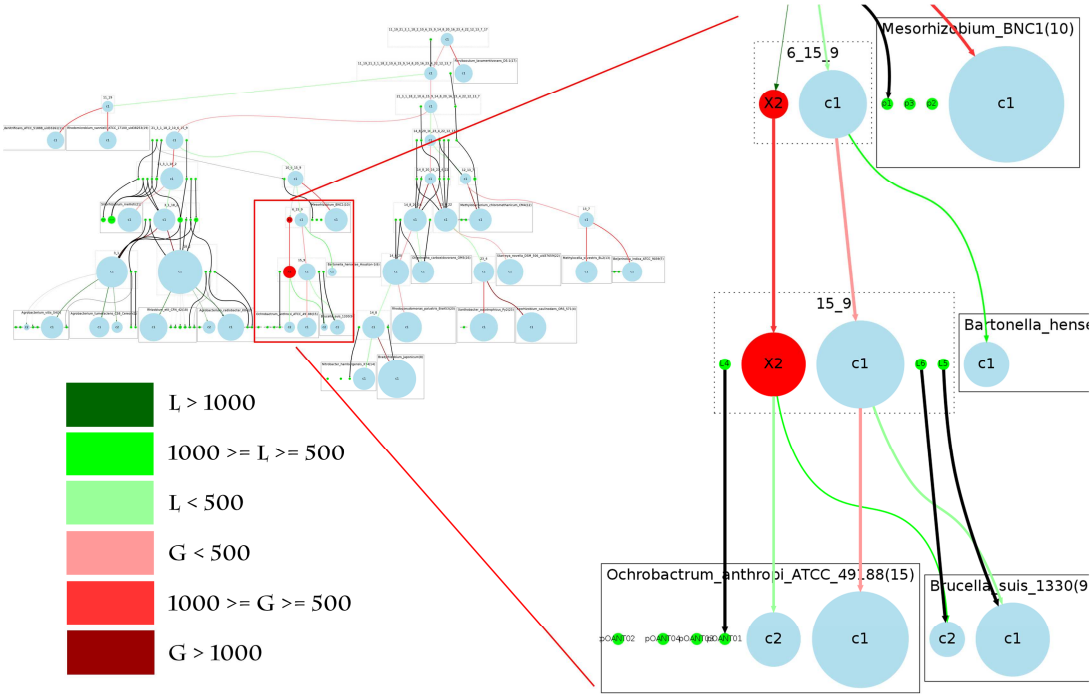

Figure 6.10. A look at the complete reconstructed evolutionary history of the Rhizobiales group. *Circles within nondotted rectangles represent the input genomes,*

while circles within dotted rectangles represent ancestral genomes. Chromosomes are shown in light blue, plasmids in green. The reconstructed secondary chromosomes are shown in red. Circle size corresponds to the number of genes it contains, except that small plasmids are kept at the same size. Edge width corresponds to the strength of the inheritance relationships between replicons, and color (given in the figure key) shows the gain (G) or loss (L) of genes on chromosomes. Edges connected with plasmids are all marked black. A part of the overview is zoomed in to give readable details. A file containing a fully zoomable version of this figure is available in the supplemental material.

Evaluation with operon structure information

To obtain a measurable evaluation for non-simulated data, we used operon structure information [61] to validate the reconstructed gene runs for ancestral genomes. Conveniently, the operon information is stored in the format of gene pairs in [63], which will be referred to as operon gene pairs from now on.

We assume that the percentage of operon gene pairs out of all gene pairs in the reconstructed gene runs in an ancestral genome should be similar to the percentage in the gene runs in input genomes. We also expect that the percentage will increase as the reconstruction approaches the root of the tree, since highly conserved gene pairs are more likely to be reconstructed as present in the ancestral genomes. Of the 23 Rhizobiales genomes, 19 of them have operon gene pair predictions. We divided all operon gene pairs into three groups by the number of genomes they occur in. The highly conserved (HC) group contains 184 operon gene pairs occurring in 10 or more genomes, the moderately

conserved (MC) group contains 688 operon gene pairs occurring in between 6 and 9 genomes and the less conserved (LC) group contains 3792 operon gene pairs occurring in between 2 and 5 genomes. Operon gene pairs occurring in only a single genome are not considered due to the lack of conservation. The overview of the occurrence of all operon gene pairs is shown in Figure 6.11.

The number of operon gene pairs in MC and LC from the ancestral genomes is similar to the correspondence from the extant genome. However, as we expected, the number in HC from ancestral genomes is significantly higher (two tailed t-test, *p* value = 2.693E-05). The number of HC operon gene pairs is also correlated (r=0.81, correlation test) with the level of the ancestor, which is defined as the number of edges the ancestor species is away from the extant species. Results support both of our expectations.

We also examined the status of the reconstructed gene pairs and gene runs in the LCA of *Rhizobiales* in terms of operon gene pair support. We say that a gene run is supported if 60% or more of the gene pairs involved are operon gene pairs. By using this criterion, 228 out of the total 305 reconstructed gene runs in the LCA genome are supported. Furthermore, out of the total 1016 reconstructed neighboring gene pairs in the LCA genome, 770 are operon gene pairs, which also support our expectation.

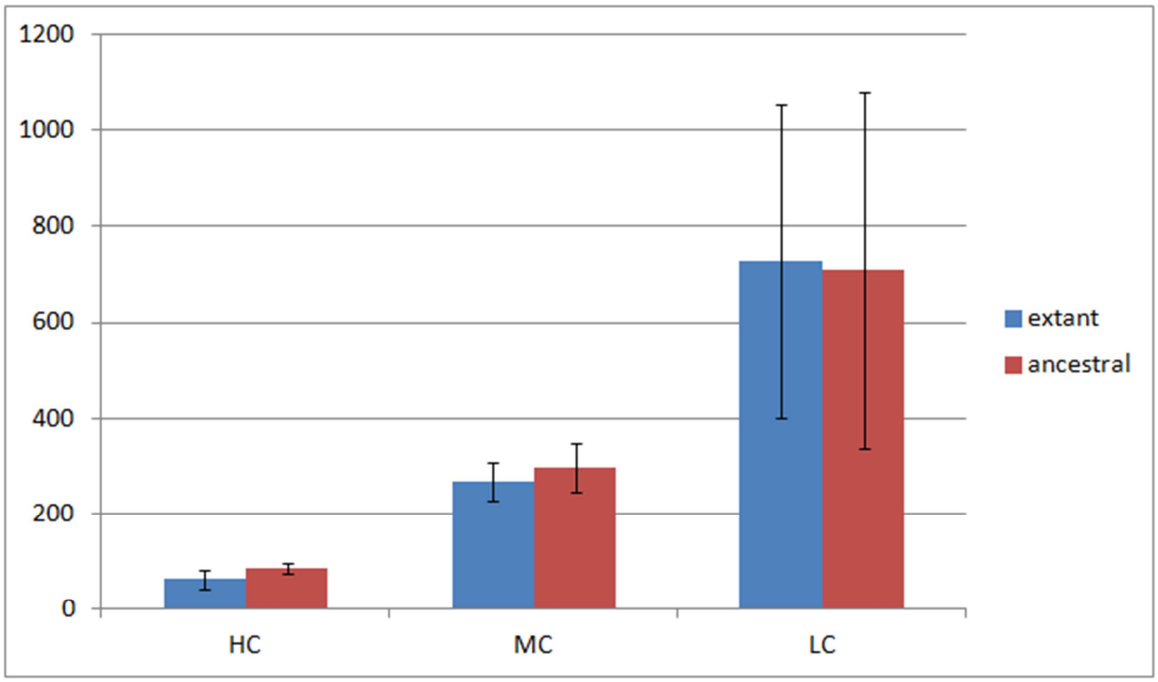

Figure 6.11. Operon gene pairs quantity comparison between extant and ancestral genomes pairs involved are operon gene pairs.

By using this criterion, 228 out of the total 305 reconstructed gene runs in the LCA genome are supported. Furthermore, out of the total 1016 reconstructed neighboring gene pairs in the LCA genome, 770 are operon gene pairs, which also support our expectation.

Leave-one-out test

We carried out a series of leave-one-out tests to examine the stability of our reconstruction method.

We performed 22 different ancestral reconstructions of the *Rhizobiales* data set at gene pair cutoff = 0.85 and 0.9 with each one of the *Rhizobiales* genomes left out. To simplify

the analysis process, we focus on the reconstructed gene runs with at least four genes for the LCA of all *Rhizobiales* species.

For each of the selected reconstructed gene runs, we scan through all 22 leave-one-out reconstructions and determine if a similar enough gene run has also been produced, which is defined as sharing at least 80% of its genes with the original. If 18 or more (~82%) leave-one-out reconstructions produce a similar enough gene run, we mark the original *recovered*, otherwise *missed*. During the analysis of the missed gene runs, we quickly realized that many gene runs are marked missed simply because they are broken into two or more fragments in the leave-one-out reconstructions by missing only a few gene pairs. We then loosened our criteria by marking a gene run recovered even if it has been broken into several fragments as long as the longest two fragments contains at least 80% of the genes in the original. The result is shown in Table 6.7.

Table 6.7. Leave-one-out stability test result. The table shows the difference in the number of gene runs as well as percentage under different cutoffs.

Number of Fragments	1		2	
gene pair cutoff	0.85	0.9	0.85	0.9
recovered	77	103	107	118
missed	80	31	50	16
total	157	134	157	134
percentage	49.04%	76.87%	68.15%	88.06%

As we can see from the table, regardless of how many fragments we allow, reconstruction with 0.9 as gene pair cutoff achieve higher recovered percentage compared with the less

stringent 0.85 cutoff, meaning that missing one genome has less impact on the more conserved reconstruction.

It also shows that when gene-pair cutoff is set to 0.9, we recover 88% of the gene runs. This number should not be taken directly as an accuracy measure, since removing one genome from the data set will inevitably lead to the lack of information to successfully reconstruct some of the original gene runs. It should be treated as a lower bound on the accuracy in the worst case.

Comparison to previous work

We compare our results with those in Slater *et al.* [2] and Boussau *et al.* [1]. The reconstruction shown in Slater *et al.* [2] is more closely related to this study, because they also reconstructed Rhizobiales ancestors and because they attempted reconstruction of conserved blocks and replicon evolution. Boussau *et al.* [1] is a more general computational inference of gene content and functional composition of genomes, focusing on the alpha-proteobacterial genomes available at the time of that study (2004).

Using a manual reconstruction, Slater *et al.* identified a few conserved gene runs that are shared by a group of *Rhizobiales* species. We mapped the species onto our tree and compared the identified conserved gene runs in the reconstructed genome of the corresponding ancestor. After the mapping, the status of a conserved block identified by Slater *et al.* can be one of the following: 1) *identical*, meaning an identical gene run is also reconstructed in our study; 2) *extended*, meaning the mapped reconstructed gene run is longer than the original conserved gene run; 3) *fragmented*, meaning the gene run is

mapped to more than one reconstructed gene runs in our study; 4) *inconsistent*, meaning there is some difference between the conserved gene run with the reconstructed gene run in our study, and 5) *missing*, meaning it failed to be mapped to any reconstructed gene run. Out of the 31 conserved gene runs in their work, eight are identical, 13 extended, four missing, five fragmented, and one inconsistent. All cases of discrepancy, including missing, fragmented, and inconsistent, are due to the difference in the genomes used in the studies. Details can be found in the supplemental material.

The chromosomal size gain and loss have shown both agreement and difference with the reconstruction made by *Boussau et al*. For example, the genome of *S. meliloti* experienced a mild gain from its LCA with *A. tumefaciens C58*. However, the sizes of ancestral genomes are generally smaller in this study, which we suspect resulted from the stringent probability cutoff.

Upon close examination of the genome functional annotation file, we noticed that the root species for these members of the *Rhizobiales* order contains genes vital for survival, as expected. Overall, more than 500 genes are categorized as involved in metabolism in the KEGG Orthology. There are 54 genes in the A-polymerase pathway (ko03010 KEGG entry), and 24 genes in Aminoacyl-tRNA biosynthesis (ko00970). Boussau *et al.* pointed out that their reconstructed ancestor has genes for glycolysis and a complete system for aerobic respiration system. A similar result is found in this study, in that the ancestral genome contains 22 genes in the Glycolysis/Gluconeogenesis pathway, covering 18 different KEGG Orthology functional annotations.

One other prediction we can make in regard to ancestral phenotypic features is the mobility of the ancestor. The ancestor possesses 14 genes in the bacterial chemotaxis pathway and 47 genes in the flagella assembly pathway, which strongly suggests that it is capable of moving and sensing the chemical signals in the surrounding environment.

6.3 Additional Remarks

This is the first automated computational method that can systematically perform ancestral genome reconstruction at both gene and replicon scales without prior assumptions on the ancestral genome replicon architecture. It is also the first method that can reconstruct gene runs for ancestral genomes with fully resolved strand information in bacteria with functional annotation using external databases. We have also modified and improved the original NGP-based model-free method so it does not require a reference genome, reconstructs all possible conserved blocks in the situation of uncertainty, correctly handles strand information, and employs a two-step occurrence uncertainty resolution process. Based on the reconstructed genomes, REGEN can also propose possible scenarios on the evolutionary events for both gene runs and replicons along the branches in the species tree.

In the functional annotated gene runs reconstructed for the LCA of all Rhizhobiales species, we noticed a small number of genes with no function assigned. With most genes on the same gene run are annotated with similar or related functional and the fact that the genes reconstructed at the root of the tree are very likely to be functionally conserved, we propose possible functional annotation of these “unknown” genes with the functions of their neighboring genes.

One significant limitation of REGEN is that since the reconstructions are performed on identified orthologous genes and gene families generated by 3rd party programs, OrthoMCL in our case, the amount of information that can be reconstructed is directly limited by the output of these programs. For example, the genes in the repABC systems of *Agrobacterium* organisms are not reconstructed because OrthoMCL failed to group them into orthologous gene families. Theoretically, it is possible to add genes with known orthologous relationships into the reconstruction, just as the refinement module does, but it involves both necessary expertise in the species of interests and manual editing of the program's output file.

There are also several important assumptions and simplifications made by the program. First of all, the replicon reconstruction algorithm assumes that in the two child species groups sharing more genes are more likely to be on the same replicon in the ancestral genome. This could be unrealistic if some large-scale evolutionary events affected a large number of genes in an uneven fashion. Second, the system will only work with bifurcating trees by design. Third, there is no concept of time in the current project. Due to the lack of data to determine mutation rates of events at all different scales, including gene-, replicon-, and genome-scale, we decided to leave the concept of time out of the scope of the current study. Without it, we cannot determine which ancestral species actually co-existed at the same point in time. Furthermore, we cannot reconstruct evolutionary events that involved more than one ancestral species, such as horizontal gene transfer from one ancestral species to another. On the other hand, these very simplifications make this method feasible and make ancestral genome reconstruction for

about 13,000 orthologous gene families in 23 species achievable in a few hours on a regular desktop (time for all-against-all BLAST is not counted).

In summary, our research has, for the first time, made automated bacterial ancestral genome reconstruction with replicon structure possible.

A version of this chapter is under review as a research article in the journal *Genes* (Yang K, Heath LS, and Setubal JC: REGEN: Ancestral Genome Reconstruction for Bacteria, 2012). Referees have asked for modifications, and a revised version has been submitted as of this writing.

Chapter 7

Conclusion

In this dissertation, we have described a new system for ancestral genome reconstruction for prokaryotes, which we have called REGEN. Two of the components of REGEN deserved special attention and were described in separate chapters.

In Chapter 4 we described a systematic methodology to refine ortholog identification predictions generated by third party *de novo* prediction programs by combining local synteny and phylogeny.

In Chapter 5 we described the development and evaluation of the first whole genome simulator for prokaryotes, which we called PEGsim. PEGsim can simulate the evolution of prokaryotic genomes at the gene and replicon scales. We have shown that PEGsim is capable of producing data with tunable properties (such as genome size and number of replicons among extant species), mimicking observed properties of actual genomes.

In Chapter 6 we described REGEN, the first automated computational method that can systematically perform ancestral genome reconstruction at both gene and replicon scales without prior assumptions on the ancestral genome replicon architecture. We applied REGEN to simulated data produced by PEGsim and to real data from members of the Rhizobiales bacterial order. Ideas for extension of the work are outlined in the conclusion of Chapter 6.

With the continued accumulation of genome data in public repositories, including an effort to cover gaps in the phylogenetic coverage of prokaryotic species [58], we can expect that REGEN has the potential of becoming an important tool in the study of prokaryote evolution. That same accumulation should also allow refinements of PEGsim and improvements of various aspects of REGEN based on additional tests on both simulated data and real genomes.

In addition to the work presented here, the author also contributed to the following publications while doing his doctoral work:

> Mining for Meaning: Visualization Approaches to Deciphering *Arabidopsis* Stress Responses in Roots and Shoots
> Lecong Zhou, Christopher Franck, Kuan Yang, Guillaume Pilot, Lenwood S. Heath, and Ruth Grene. *OMICS*: A Journal of Integrative Biology. April 2012, 16(4): 208-228. doi:10.1089/omi.2011.0111.
>
> Next-generation phage display: integrating and comparing available molecular tools to enable cost-effective high-throughput analysis
> Emmanuel Dias-Neto, Diana Nunes, Ricardo Giordanol, Jessica Sun, Gregory Botz, Kuan Yang, Joao Setubal, Renata Pasqualini, Wadih Arap
> *PLoS One*, 2009,4(12):e8338.
>
> Performance comparison of gene family clustering methods with expert-curated gene family dataset in *Arabidopsis thaliana*.

Kuan Yang, Liqing Zhang

Planta. DOI: 10.1007/s00425-008-0748-7

Performance comparison between k-tuple distance and four model-based distances in phylogenetic tree reconstruction

Kuan Yang, Liqing Zhang

Nucleic Acids Research, 2008, Vol. 36, No. 5 e33

REFERENCES

1. Boussau B, Karlberg EO, Frank AC, Legault BA, Andersson SG: **Computational inference of scenarios for alpha-proteobacterial genome evolution**. *Proc Natl Acad Sci U S A* 2004, **101**(26):9722-9727.
2. Slater SC, Goldman BS, Goodner B, Setubal JC, Farrand SK, Nester EW, Burr TJ, Banta L, Dickerman AW, Paulsen I *et al*: **Genome Sequences of Three Agrobacterium Biovars Help Elucidate the Evolution of Multi-Chromosome Genomes in Bacteria**. *J Bacteriol* 2009, **191**:2501-2511.
3. Sankoff D, Rousseau P: **Locating the vertices of a steiner tree in an arbitrary metric space**. *Mathematical Programming* 1975, **9**:240-246.
4. Fitch WM: **Toward defining the course of evolution : minimum change for a specific tree topology**. *Systematic Zoology* 1971, **20**:406-416.
5. Bhutkar A, Gelbart WM, Smith TF: **Inferring genome-scale rearrangement phylogeny and ancestral gene order: a Drosophila case study**. *Genome Biology* 2007, **8**(R236).
6. Pe'er I, Shamir R: **The median problems for breakpoints are NP-complete**. *Electronic Colloquium on Computational Complexity* 1998:71.
7. Moret BME, Tang J, Wang L-S, Warnow T: **Steps toward accurate reconstructions of phylogenies from gene-order data**. *Journal of Computer and System Sciences* 2002, **65**(3):508-525
8. Caprara A: **Formulations and hardness of multiple sorting by reversals**. In: *In proceedings of the 3rd International Conference of Computational Molecular Biology: 1999; Lyon, France*. ACM Press: 84-93.
9. Moret BMET, J. ; Warnow, T. ; Gascuel, O.: **Reconstructing phylogenies from gene-content and gene-order data**: Oxford University Press; 2005.
10. Bourque G, Tesler G, Pevzner PA: **The convergence of cytogenetics and rearrangement-based models for ancestral genome reconstruction**. *Genome Res* 2006, **16**(3):311-313.
11. Eriksen N: **Reversal and transposition medians**. *Theoretical Computer Science* 2007, **374**:111-126.
12. Fitch WM: **Distinguishing homologous from analogous proteins**. *Syst Zool* 1970, **19**(99-106).
13. Heijden RTvd, Snel B, Noort Bv, Huynen MA: **Orthology prediction at scalable resolution by phylogenetic tree analysis**. *BMC Bioinformatics* 2007, **8**(83).
14. Snel B, Huynen MA: **Quantifying Modularity in the Evolution of Biomolecular Systems**. *Genome Research* 2004, **14**(3):391-397.
15. Stone EA, Sidow A: **Physicochemical constraint violation by missense substitutions mediates impairment of protein function and disease severity**. *Genome Research* 2005, **15**(7):978-986.
16. Jothi R, Zotenko E, Tasneem A, Przytycka TM: **COCO-CL: hierarchical clustering of homology relations based on evolutionary correlations**. *Bioinformatics* 2006, **22**(7):779-788.
17. Storm CEV, Sonnhammer ELL: **Automated ortholog inference from phylogenetic trees and calculation of rothology reliability**. *Bioinformatics* 2002, **18**(92-99).

18. Dufayard J-F, Duret L, Penel S, Gouy M, Rechenmann F, Perrière G: **Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases**. *Bioinformatics* 2005, **21**(11):2596-3603.
19. Zmasek CM, Eddy SR: **RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs**. *BMC Bioinformatics* 2002, **3**(14).
20. Koski LB, Golding GB: **The Closest BLAST Hit Is Often Not the Nearest Neighbor** *Journal of Molecular Evolution* 2004, **52**(6):1432-1432.
21. Remm M, Storm CEV, Sonnhammer ELL: **Automatic clustering of orthologs and inparalogs from pairwise species comparisons**. *Journal of Molecular Biology* 2001, **214**(5):1041-1052.
22. Li L, Jr. CJS, Roos DS: **OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes**. *Genome Research* 2003, **13**(9):2178-2189.
23. Hubbard TJP, Aken BL, Beal K, Ballester B, Caccamo M, Chen Y, Clarke L, Coates G, Cunningham F, Cutts T *et al*: **Ensembl 2007**. *Nucleic Acids Res* 2007, **V35**:610-617.
24. Wheeler DL, Barrett T, Benson DA, Bryant SH, Canese K, Church DM, DiCuccio M, Edgar R, Federhen S, Helmberg W *et al*: **Database resources of the National Center for Biotechnology Information** *Nucleic Acids Res* 2005, **33**:D39-D45.
25. Kuzniar A, Ham RCHJv, Pongor Sn, Leunissen JAM: **The quest for orthologs: finding the corresponding gene across genomes**. *Trends in Genetics* 2008, **24**(11):539-551.
26. Hulsen T, Huynen MA, Vlieg Jd, Groenen PM: **Benchmarking ortholog identification methods using functional genomics data**. *Genome Biology* 2006, **7**(4):R31.
27. Lemoine F, Lespinet O, Labedan B: **Assessing the evolutionary rate of positional orthologous genes in prokaryotes using synteny data**. *BMC Evolutionary Biology* 2007, **7**(237).
28. Yang K, Setubal JC: **A Whole Genome Simulator of Prokaryote Genome Evolution**. In: *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine: 2011; Chicago, Illinois*. 2147885: ACM: 508-510.
29. Cartwright RA: **DNA assembly with gaps (Dawg): simulating sequence evolution**. *Bioinformatics* 2005, **21**(Suppl 3):iii31-iii38.
30. Beiko RG, Charlebois RL: **A simulation test bed for hypotheses of genome evolution**. *Bioinformatics* 2007, **23**(7):825-831.
31. Varadarajan A, Bradley RK, Holmes IH: **Tools for simulating evolution of aligned genomic regions with integrated parameter estimation**. *Genome Biology* 2008, **9**(R147).
32. Cunningham CW: **Some Limitations of Ancestral Character-State Reconstruction When Testing Evolutionary Hypotheses**. *Systematic Biology* 1999, **48**(3):665-674.

33. Cunningham CW, Omland KE, Oakley TH: **Reconstructing ancestral character states: a critical reappraisal**. *Trends in Ecology & Evolution* 1998, **13**(9):361-366.
34. Ronquist F: **Bayesian inference of character evolution**. *Trends in Ecology & Evolution* 2004, **19**(9):475-481.
35. Vanderpoorten A, Goffinet B: **Mapping Uncertainty and Phylogenetic Uncertainty in Ancestral Character State Reconstruction: An Example in the Moss Genus Brachytheciastrum**. *Systematic Biology* 2006, **55**(6):957-971.
36. Koonin EV: **Orthologs, Paralogs, and Evolutionary Genomics**. *Annu Rev Genet* 2005, **39**:309-338.
37. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs**. *Nucleic Acids Research* 1997, **25**(17):3389-3402.
38. Wu M, Eisen JA: **A simple, fast, and accurate method of phylogenomic inference. Genome Biol. 2008;9(10):R151**. *Genome Biology* 2008, **9**(R151).
39. Yang K, Setubal JC: **Homology prediction refinement and reconstruction of gene content and order of ancestral bacterial genomes**. In: *Proceedings of the First ACM International Conference on Bioinformatics and computational Biology: 2010; Niagara Falls, New York, U.S.A.*: ACM: 230-236.
40. David LA, Alm EJ: **Rapid evolutionary innovation during an Archaean genetic expansion**. *Nature* 2011, **469**:93-96.
41. Reed WJ, Hughes BD: **From gene families and genera to incomes and internet file sizes: Why power laws are so common in nature**. *Physical Review* 2002, **66**:067103.
42. Searls DB: **The language of genes**. *Nature* 2002, **420**:211-217.
43. Deák I: **Random Number Generators and Simulation**. New York: State Mutual Book & Periodical Service; 1990.
44. Bratley P, Fox BL, Schrage LE: **A guide to Simulation**. New York: Springer-Verlag; 1996.
45. Kowalski J, Waga W, Zawierta M, Cebrat S: **Phase Transition in the Genome Evolution Favors Nonrandom Distribution of Genes on Chromosomes**. *International Journal of Modern Physics C* 2009, **20**(08):1299-1309.
46. Ryabov Y, Gribskov M: **Spontaneous symmetry breaking in genome evolution**. *Nucleic Acids Research* 2008, **36**(8):2756-2763.
47. Zdobnov EM, Bork P: **Quantification of insect genome divergence**. *Trends in Genetics* 2007, **23**(1):16-20.
48. Souciet J-L, Dujon B, Gaillardin C, Johnston M, Baret PV, Cliften P, Sherman DJ, Weissenbach J, Westhof E, Wincker P *et al*: **Comparative genomics of protoploid Saccharomycetaceae**. *Genome Research* 2009, **19**(10):1696-1709.
49. de Queiroz A, Gatesy J: **The supermatrix approach to systematics**. *Trends in ecology & evolution (Personal edition)* 2007, **22**(1):34-41.
50. Edgar RC: **MUSCLE: multiple sequence alignment with high accuracy and high throughput**. *Nucleic Acids Res* 2004, **32**(5):1792-1797.
51. Castresana J: **Selection of Conserved Blocks from Multiple Alignments for Their Use in Phylogenetic Analysis**. *Molecular Biology and Evolution* 2000, **17**(4):540-552.

52. Stamatakis A, Hoover P, Rougemont J: **A rapid bootstrap algorithm for the RAxML Web servers**. *Syst Biol* 2008, **57**(5):758-771.
53. Buschbom J, Barker D: **Evolutionary History of Vegetative Reproduction in Porpidia s.l. (Lichen-Forming Ascomycota)**. *Systematic Biology* 2006, **55**(3):471-484.
54. Kruskal J: **On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem**. *Proceedings of the American Mathematical Society* 1956, **7**(1):48-50.
55. Cormen T, Leiserson C, Rivest R, Stein C: **Introduction to Algorithms**: McGraw-Hill Science / Engineering / Math; 2003.
56. Harrison PW, Lower RPJ, Kim NKD, Young JPW: **Introducing the bacterial 'chromid': not a chromosome, not a plasmid**. *Trends in Microbiology* 2010, **18**(4):141-148.
57. Kanehisa M, Goto S: **KEGG: Kyoto Encyclopedia of Genes and Genomes**. *Nucleic Acids Research* 2000, **28**(1):27-30.
58. Wu D, Hugenholtz P, Mavromatis K, Pukall R, Dalin E, Ivanova NN, Kunin V, Goodwin L, Wu M, Tindall BJ *et al*: **A phylogeny-driven genomic encyclopaedia of Bacteria and Archaea**. *Nature* 2009, **462**:1056-1060.
59. Kolaczkowski B, Thornton JW: **Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous**. *Nature* 2004, **431**(7011):980-984.
60. Ellson J, Gansner E, Koutsofios E, North S, Woodhull G: **Graphviz - Open Source Graph Drawing Tools**. *Graph Drawing* 2001:483-484.
61. Price MN, Huang KH, Alm EJ, Arkin AP: **A novel method for accurate operon prediction in all sequenced prokaryotes**. *Nucleic Acids Research* 2005, **33**(3):880-892.