Chapter 5
Integration of Evolutionary Biology Concepts for Functional Annotation and Automation of Complex Research in Evolution: The Multi-Agent Software System DAGOBAH

Philippe Gouret, Julien Paganini, Jacques Dainat, Dorra Louati, Elodie Darbo, Pierre Pontarotti, and Anthony Levasseur

Abstract Various strategies have been proposed for predicting protein function. They are derived from the classical homology-based approaches and emerging alternative approaches taking into account gene history in the framework of phylogenetic comparative methods. The growing numbers of available genome sequences and data require bioinformatics tools, in which methodological approaches are set according to the biological issues to be addressed. Much effort has already been devoted to integrating evolutionary biology into bioinformatics tools; e.g., homology-based functional annotation has been successfully integrated in a pipeline-assisted method. In addition, new concepts based on correlation of evolutionary events are emerging. For example, two independent events (e.g., systematic loss of specific genes) that happen repetitively can therefore be functionally linked. However, correlated gene profiles, also called "contextual annotation," makes use of different bioinformatics resources based on multi-agent development. In this chapter, we describe evolutionary concepts and bioinformatics approaches proposed for future functional inference.

P. Gouret • J. Paganini • J. Dainat • E. Darbo • P. Pontarotti
UMR6632, Evolutionary Biology and Modeling, Université de Provence, 3 place Victor Hugo, 13331 Marseille, France
e-mail: philippe.gouret@univ-provence.fr

D. Louati
UMR6632, Evolutionary Biology and Modeling, Université de Provence, 3 place Victor Hugo, 13331 Marseille, France

(LAMSIN-IRD) ENIT, Ecole Nationale d'Ingénieurs de Tunis BP 37, Le Belvèdere 1002-Tunis, Tunisia

A. Levasseur
INRA, UMR1163 de Biotechnologie des Champignons Filamenteux, IFR86-BAIM, Universités de Provence et de la Méditerranée, ESIL, 163 avenue de Luminy, CP 925, 13288 Marseille Cedex 09, France

Universités Aix-Marseille 1 et 2, UMR1163, 163 avenue de Luminy, CP925, 13288 Marseille Cedex 09, France

P. Pontarotti (ed.), *Evolutionary Biology – Concepts, Biodiversity, Macroevolution and Genome Evolution*, DOI 10.1007/978-3-642-20763-1_5,

5.1 Functional Annotation Strategies: Current and Future Approaches

5.1.1 Homology-Based Functional Annotation

Eisen was the first to conceptually rationalize phylogenetic methods to improve the accuracy of functional predictions. In 1998, he proposed a phylogenetic prediction of gene function and compared it to similarity-based functional prediction methods (Eisen 1998). In this work, all known functions on a phylogenetic tree were overlaid. The prediction task could then be split into two steps. In the first step, the tree could be used to decipher orthology and paralogy relationships. Most of the reports based on evolutionary biology methods used ortholog information to transfer functional annotation (see Gouret et al. 2005 and Danchin et al. 2007). Functional assignment could be performed for uncharacterized proteins only if the function of an ortholog was known (and if a similar function was evidenced for all characterized orthologs). Ideally, functional inference should be carried out for experimentally validated orthologs. Bibliographic analysis indicates that orthologs are more likely to keep a similar function than paralogs (e.g., Collette et al. 2003). Theoretically, after duplication, one of the copies is lost, or both duplicates undergo subfunctionalization, or one of the duplicates evolves toward a new function (Force et al. 1999). However, Studer has challenged this assumption, as orthologs and paralogs could have comparable mechanisms of divergence (Studer and Robinson-Rechavi 2009). Different and more complex fates of duplicates could also be evidenced (for a review, see Levasseur and Pontarotti 2011).

In the second step, parsimony reconstruction or alternative reconstructive propagation methods could be used to assign functions of uncharacterized genes by identifying the evolutionary scenario that requires the fewest functional changes over time. Inference of ancestral state on phylogenetic tree requires that character mapping be accurate. Uncertainty about trees and mapping is therefore counterbalanced by introducing Bayesian statistical methods, taking into account this inherent error parameter (Ronquist 2004).

To the best of our knowledge, the first report using both approaches was integrated in the work of Engelhardt et al. (2005). The authors constructed a model of molecular function evolution to infer function in a phylogenetic tree. The model takes into account evidence of varying quality and computes a posterior probability for every possible molecular function for each protein in the phylogeny. Different hypotheses were included in the strategy, i.e., each molecular function may evolve from any other function, and a protein's function may evolve more rapidly after duplication events than after speciation events (Engelhardt et al. 2005). Branch length and duplication are integrated in the methodological approach. In brief, methods may be summarized as propagating functional information from leaves to the root of the phylogeny and then propagating back out to the leaves of the phylogeny, based on the probabilistic model of function evolution.

Homology-based functional annotation is summarized in Fig. 5.1.

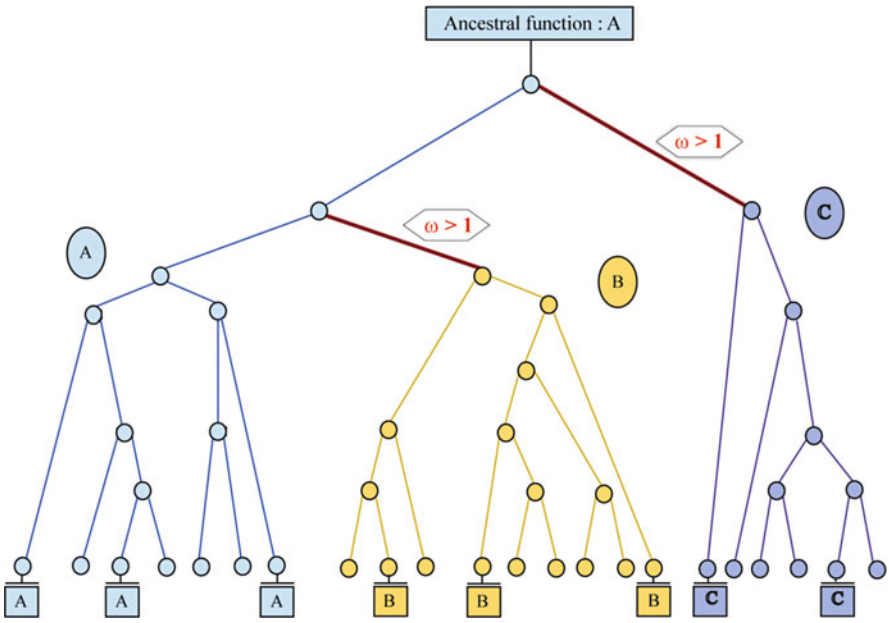

Fig. 5.1 Homology-based functional annotation. Functionally annotated leaves are labeled, respectively, as function A (*blue*), B (*yellow*), and C (*dark blue*). Putative function of non-annotated leaves is inferred after ancestral reconstruction based on propagation of functional information from leaves to the root of the phylogeny. Red branches: evolutionary and functional shift (using $\omega = dN/dS > 1$, i.e., Darwinian selection). (Adapted from Levasseur and Pontarotti 2008)

5.1.2 *Strengthening Functional Annotation: Integration of Correlative Approaches*

Functional prediction using "contextual information" is tricky because of (i) technical difficulty in detecting occurrence profiling and (ii) statistical methods required to correlate and infer function accurately. Co-occurrence and correlated gene profiles could result from phylogenetic inheritance among closely related species. Alternatively, co-occurrence could also result from individual adaptive functions, for instance when genes appear or are lost independently in several distinct lineages (Barker and Pagel 2005). Thus the probability of functional linkage between genes is proportional to the number of multiple independent phylogenetic events. A simplified example of co-occurrence and functional links is depicted in Fig. 5.2. Unlike the overall counting of presence or absence of genes, phylogenetic methods enable us to investigate ancestral states and decipher independent multiple evolutionary events.

Different methods for occurrence profiling have already been proposed, mainly on the basis of the parsimony principle and maximum likelihood (ML).

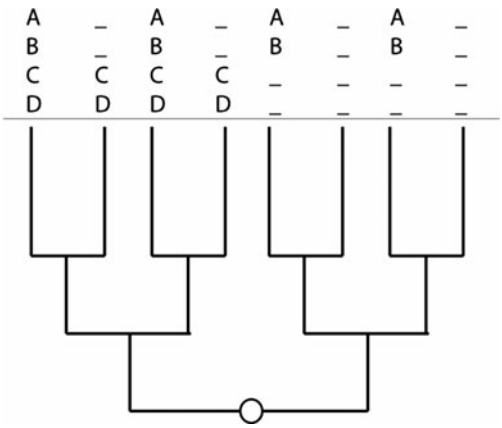

Fig. 5.2 Co-occurrence and functional link. Example of the need for comparative phylogenetic methods. Presence/absence of genes (A, B, C, D) is reported on the leaves of the phylogenetic tree. Here, multiple independent phylogenetic events of gain/loss of gene pairs (i.e., four independent events for genes A and B) are opposed to the apparent correlation arising from shared inheritance of gene pairs loss (resulting from one ancient event for genes C and D). The different steps can be summarized as follows: (i) detection of event: A is lost, (ii) convergence detection: A is lost several times, (iii) co-convergence detection: A and B are lost together several times. Subsequently, statistical tests are carried out. The function of non-annotated genes could be deduced from the correlated annotated genes

As described in the work of Barker and Pagel (2005) and Barker et al. (2007), a common pattern of presence and absence across a range of distinct genomes could be integrated as a method for detecting functionally linked proteins. Thus correlated gains and losses of genes on a phylogenetic tree of species could improve the detection of functionally linked pairs of proteins, compared with the original across-species methods from Pellegrini et al. (1999). Several phylogenetic methods were compared in their work to evaluate the accuracy of their method. Methods were based on either Dollo parsimony (Farris 1977) or ML, including a general model, but also using a constrained model in which the rate of gain of genes is not estimated from the data, but set at a low value. The fixed value of the ML should model gene content evolution better, by preventing the modeling of multiple gains of the same gene in different parts of the phylogeny. In the parsimony case, the reconstructed ancestral states could be very uncertain and parsimony could be applied when rates of changes are rather low. Note that parsimony intervals are proposed to account for the uncertainty of the parsimony methods. For instance, Zhou et al. proposed a dynamic programming algorithm to calculate such parsimony intervals. The best 100 suboptimal ancestral states were determined, and the authors compared the number of correlated events, while allowing for the degree of suboptimality of the reconstructions (Zhou et al. 2006). By contrast, ML accounts for the branch length and uncertainty of topology in the tree, and the estimate of the likelihood values is an independent parameter (i.e., corresponding to all ancestral state possibilities). The authors conclude that all the phylogenetic methods except

unconstrained ML achieved higher specificity than the across-species approach (ML model being capable of greater accuracy and sensitivity than a Dollo parsimony-based approach) (Barker et al. 2007).

5.1.3 *Toward Reliable Global Functional Annotation: The Need for Bioinformatics*

Bioinformatics has unlocked vast amounts of genomic data and developed software applications based on increasingly powerful mathematical algorithms – which themselves produce large volumes of results –, but the amounts of data involved simply cannot be interpreted with any real depth using statistical correlations. We therefore need to develop smart software systems able to support researchers in their efforts, which means systems automatically handling the major routine component of their *in silico* research protocols, and helping analysts interpret the huge volumes of results generated. Such smart software systems could ease the most burdensome part of the workload, leaving researchers to channel their energy into the "sharp end" of their research.

In early 2002, evolutionary biologists were handling vast quantities of biological data made available through the Internet, and running an array of software tools based on probabilistic algorithms working on these data or on data derived from other mathematical tools. The models associated with these tools were all task-specific – sequence similarity, gene prediction, phylogenetic tree-building, and so on. However, they never integrated a large number of concepts employed in biological knowledge and reasoning into a single, integrative software solution. Hence individually, they were unable to answer complex questions posed by biologists or to verify their hypotheses. Consequently, we had to automatically chain mathematical computations through what bioinformaticians call pipelines.

According to the functional annotation strategies described above, homology and correlative approaches were integrated into specific bioinformatics platforms.

A bioinformatics strategy designed for homology-based functional annotation was first implemented by creating FIGENIX (Gouret et al. 2005). FIGENIX is a Java (java.sun.com) platform that automates simple pipeline schemes, such as basic phylogenetic tree-building from a protein sequence by (i) similarity searching against protein databases, (ii) simple filtering, (iii) alignment, and (iv) tree computation. Mathematical tool chaining, through this first version of FIGENIX or any of the pipeline systems available at the time, was unable to completely automate a process: this meant that biologists still had to intervene between computation phases to verify, correct, and synthesize data output from the mathematical tools and guide the workflow to the relevant part of the pipeline. The only way to resolve this automation issue was to introduce an expert system (with Prolog language; Warren et al. 1977) into FIGENIX to model a part of biologists' knowledge and thus act as a human scientist as and when necessary. By introducing specific logical rules in the expert system, a pipeline was created and was dedicated to gene

predictions *via* an approach combining ab initio predictions and homology through a lab method. Tested against a known benchmark, the pipeline clearly proved successful. A complex phylogeny pipeline with 50 steps and a lot of expertise modeling was designed. The first version was stabilized in late 2003, and has since enabled the laboratory and its collaborators to produce thousands of phylogenic trees from protein queries. These trees form the basis of our evolutionary research. This pipeline, along with others, was intensively used on laboratory projects, generating several published papers (Danchin et al. 2004, 2006, 2007; Paillisson et al. 2007; Levasseur et al. 2006, 2010). It continued to undergo improvements and enhancements, with upgrades including automatic detection of orthologs in the final process-synthesized tree by online recovery of functional data associated with these orthologs (GO (Ashburner et al. 2000), MGI (www.informatics.jax.org), NCBI (www.ncbi.nlm.nih.gov)), and EST integration (Balandraud et al. 2005). Part of the software developed, called PhyloPattern, emerged as a crucial independent component (Gouret et al. 2009). The aim of this tool was to reproduce human reading of phylogenetic trees, i.e., phylogenetic tree annotation and pattern recognition. Inside the phylogeny pipeline, this tool is used to detect incongruence or isolate specific subtrees, from which biases are then corrected. PhyloPattern now makes it possible to detect events in the history of species, genes, or any other characteristic (from domain to function and further), as well as highlighting artifacts in the phylogenetic trees. We are continuing to improve PhyloPattern as a free open-source JAVA/Prolog API.

5.2 From Pipelines to Multi-Agent Strategies

In 2005, it became clear that the "pipeline approach," even with the controlling expertise introduced, remained limited to computation processes. In addition, functional annotation using the correlative approaches strategy required flexible and more sophisticated data processing architecture. Computation processes are essential, but are not really able to resolve complex tasks of interest to the laboratory, such as automatically highlighting genetic events in the human genome and detecting convergences and co-convergences among these events. Any solution to these issues needs to be driven by expertise through parallel and more "intelligent" processes than the rigid, deterministic pipelines. We also note that the "pipeline approach" does not extend to establishing an explicitly described semantic universe that would allow accurate meta descriptions of data. It thus remains impossible to raise the abstraction level of software tasks, and interfacing them with other software systems is not natural.

Integration of correlated gene profiles for functional annotation requires a three-step process: (i) specific detection of all evolutionary events, (ii) correlation using phylogenetic comparative methods leading to a compelling statistical results, and (iii) deducing the function of non-annotated genes from the correlated annotated genes.

5.3 Technical System Specifications

Accordingly, a new software system was conceived and is able to implement complete automation of actual full research via bottom-up (from biological data) strategies specified by the laboratory, rather than "just" complex computation workflows. We opted for the following research strategy: (i) working from known or computed features to find evidence for generating new hypotheses, (ii) attempting to verify hypotheses to transform them into features, (iii) correlating verified features to deduce new features, and so on. A set of characteristic specifications was drawn up:

- The treatments had to be flexible, modular, and parallelized.
- The strategies for identifying and verifying the facts had to be led by expertise.
- Communication with external software systems (online databases, web services) should systematically gather the relevant results produced by these platforms, such as Ensembl (Hubbard et al. 2009), NCBI, String (Szklarczyk et al. 2011), and ArrayExpress (Parkinson et al. 2011).
- The results had to be placed in an accurately described semantic universe that was not redundant but interfaced with data from external systems.
- Some modules had to work together and communicate directly, while others, such as modules for intelligent correlations of events, had to work in stand-alone mode directly on the mass of results produced by the full set of modules.
- The modules had also be able to work at different times.
- The system had to be resistant to failure; as such, very costly computational treatments should have to be run only once.

5.4 Technical State of the Art

The field of biology now has a number of software tools, approaches, standards, and publications that could be recycled for our needs. The type of system targeted here required establishing an integrated data model, placed between structured biological data (e.g., genomic databases) or unstructured data (publications) located inside or outside the laboratory, and the research strategies desired by laboratory researchers. Software systems clearly have to work with large-scale data banks, but what is most important now is to work with different kinds of data, many of which are not a direct representation of biological objects but are more abstract concepts.

We could therefore rule out relational database management systems, which are not powerful enough or flexible enough to describe semantics in biology. Some recently developed software tools such as the alignment expert system ALEXSYS (Aniba et al. 2009) are based on the UIMA framework (http://sourceforge.net/projects/uima-framework/), which offers a powerful architecture and is well-suited to the introduction of a virtual model on unstructured data, i.e., building meta-information from artifacts such as scientific publications (also see DiscoveryLink

(Hass et al. 2001) or BioMOBY (Wilkinson and Links 2002)). We are more focused on trying to directly model actual genomics or evolutionary concepts. Also, the UIMA approach is only "object-oriented," and we believe that this kind of modeling architecture is not rich enough to integrate the complexity of biological paradigms, especially compared with approaches based on mathematical first-order logic ontology techniques such as Description Logic (http://dl.kr.org/). The W3C-standardized OWL language (http://www.w3.org/TR/owl-ref/) is an XML representation of DL. Initially applied to the semantic web, it is fast becoming a standard for ontology modeling. In DL, relations between classes are not limited to aggregation or inheritance links but can be formalized with logical formulae. However, we note that DL does not integrate concepts of inductive, temporal, or fuzzy logic, which in the long term could direct the natural extension of our systems.

Biology now has many ontologies (e.g., NCI Cancer Ontology: http://www.mindswap.org/2003/CancerOntology/). Some are defined in OWL but to our knowledge, none computationally exploit the descriptive capacity of description logic (DL). This situation is surely set to change. We note the existence of relational ontology (Smith et al. 2005), placed between "object" modeling and DL modeling, which attempts to standardize relations in biological ontologies. This point will be revisited below. There appears to be a continuing dichotomy between the activity of defining ontologies, considered as vocabularies by many biologists, and the establishment of DL-based software and databases within and between laboratories or institutes. We believe that this dichotomy is an error, as it has very adverse repercussions, such as poor software systems and bad interoperability.

As stated above, to fully automate *in silico* research strategies, the type of system we are targeting has to be less rigid and deterministic than pipelines. A natural candidate solution would be multi-agent systems. In bioinformatics, these systems are used essentially to model and simulate biological networks (reactive agents), although they are also used to parallelize mathematical computations through agents with very fine granularity. They are rarely employed for building integrative applications where "smart" agents work with biological information. Nevertheless, like the FIPA institute (http://www.fipa.org/), we are convinced that this kind of architecture built from cognitive agents (with large granularity) communicating inside an ontological semantic universe can be applied to bioinformatics automation. The JADE software framework (http://jade.tilab.com/) is a Java implementation of FIPA specifications. At our lab, we used JADE to develop a first prototype multi-agent system named CASSIOPE (Rascol et al. 2009), dedicated to highlighting conserved synteny.

Recently, eHive emerged from EBI as a new workflow system (Severin et al. 2010). It is built as a multi-agent "blackboard" architecture. Here, the blackboard, i.e., the communication area between agents, is reduced to chaining rules between agents. Thus the tasks produced by the system are driven by predefined functional relations between agents and not by the autonomous interpretation, by agents, of the data resulting from other agents' work. The Ehive blackboard database has a rigid structure with no data modeling. Also, agents' source code is written with the Perl

language, which albeit very widely used in bioinformatics remains very poor in expertise and knowledge modeling.

As stated earlier, we are seeking to deploy expertise-driven research strategies, which means that all agents need to be built with expert-system architectures. Rule engines do exist – one example is Jess (www.jessrules.com) – but it would be preferable to write our own engine in Prolog language to reap the benefit of tools we developed previously, especially PhyloPattern. After years of hands-on experience, we can confirm that the Prolog language is very well-suited to bioinformatics. Its benefits for the target system include: (i) a natural capacity to generate all the solutions for a question, (ii) easy and native manipulations of lists and tree structures, which are intensively used in bioinformatics data, (iii) development of expert systems in backward- and/or forward-chaining mode (verification and/or production of facts), (iv) formalisms (e.g., ontological relations) representable directly in the language's syntax, (v) brevity and simplicity of knowledge descriptions, and (vi) interpreted language that strengthens the experimental aspect of certain developments.

5.5 System Architecture

Our system was called DAGOBAH. It is shaped as a multi-agent software (see Fig. 5.3), with a voluntarily hybrid model summing of a model called "Belief Desire Intention" with a model called "Blackboard" (Ferber 1995). The BDI model is suitable for cognitive agents with high granularity and therefore high "intelligence." In the BDI model, agents have a plan formed for our purposes by logical rules. This highly flexible rule system is used by each agent to implement a specific strategy, but can also be used as a traditional expert system to produce high-level facts deduced from simpler facts. For example, an agent capable of sifting through actions to detect several equally probable genetic events from a phylogenetic tree will be able to retain only one event, through a set of logical rules associated with a set of criteria.

The semantics for one rule is defined as follows:

- $Action_1 \ldots Action_k$
  $ConditionFact_1 \ldots ConditionFact_n \rightarrow ConclusionFact_1 \ldots ConclusionFact_m$
  $ToBeRemovedFact_1 \ldots ToBeRemovedFact_z$

The meaning is "if all condition facts (n) are known by the agent ($\subset$ Belief) and if at least one of the conclusion facts (m) is not present and if the agent is capable of achieving all actions (k) ($\subset$ Intention) successfully, then all conclusions (m) ($\subset$ Desire) are considered truthful, and all indicated facts (z) are removed from the agent's knowledge."

Here is an example rule, used in the DAGOBAH agent dedicated to searching for domain architecture events. We suppose that for a specific protein with the domain architecture A-B-C, DAGOBAH detects an event that produced the B-C part of the architecture by analyzing the phylogenetic tree of domain B, and we suppose

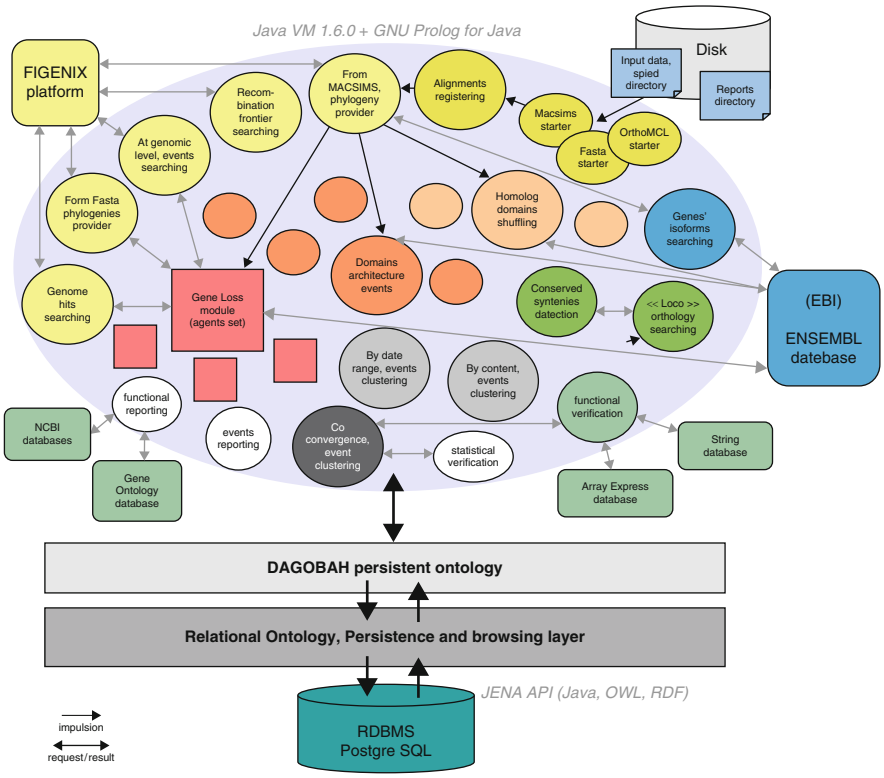

Fig. 5.3 DAGOBAH multi-agents system architecture. All agents (*disks*) or modules (*squares*) (set of agents) that compose DAGOBAH are contained in the large blue ovoid. Around it are displayed the external software systems interacting with the agents by the network. At the bottom of the scheme is shown the ontological database, containing the biological results produced and shared by the agents

that DAGOBAH hesitates between identifying the event as a shuffling or a gain. A simple rule, if it is applicable, allows DAGOBAH to definitely assert there is a gain (see Fig. 5.4):

- *verify_similarity_of_signal_between(P1, P2, [B, C])*
  event_found_under_ancestral_node(N),
  apomorphic_chosen_protein(P1, [A, B, C]), → *gain_event_found(N, P1, [C])*
  plesiomorphic_chosen_protein(P2, [A, B])
  event_found_under_ancestral_node(N)

The "Blackboard" model introduces an area of information shared by agents, i.e., any important result produced by an agent is placed on the blackboard. The blackboard architectural model chosen in DAGOBAH is defined as a persistent ontology (an ontological database) representing the semantic universe in which the agents work. These results are used by other agents, unless they are forced to

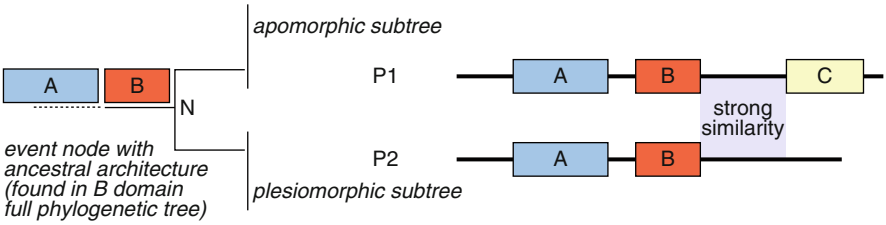

Fig. 5.4 A virtual example for a domains architecture event. Here again event is confirmed because the genomic signal between domains B and C on the apomorphic sequence is strongly conserved after domain B on the plesiomorphic sequence

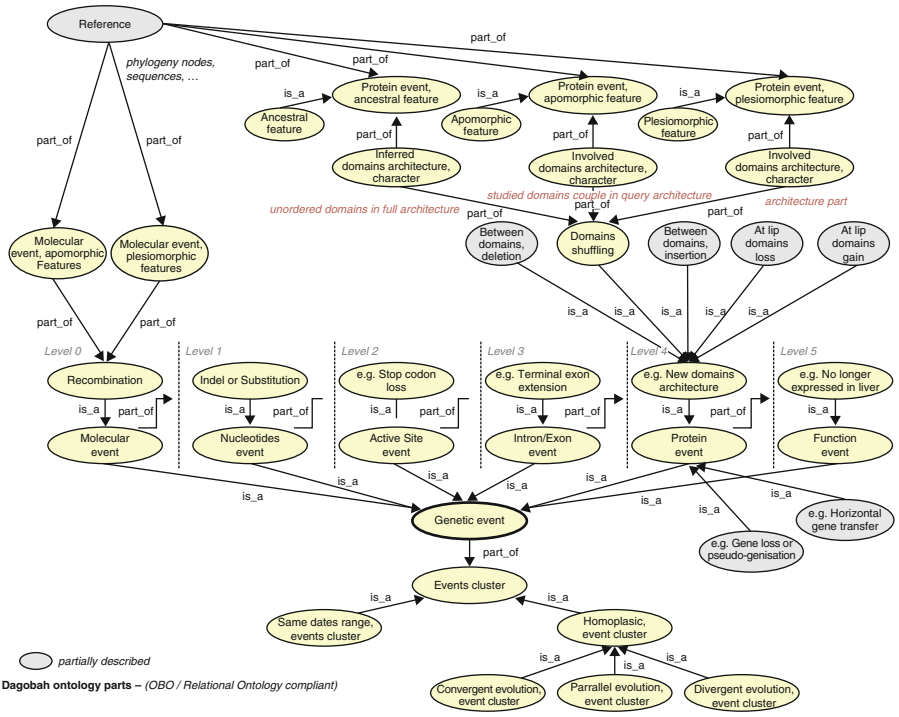

Fig. 5.5 The core of DAGOBAH ontology. Some genetic event classes laid out by their reading level are presented. As an example, we give all the classes participating in a nonhomologous domain shuffling event, induced by a recombination event. Clustering classes are also displayed with their inheritance relationships

explicitly and systematically exchange them. Figure 5.5 illustrates the main parts of the DAGOBAH ontology. Genetic event classes are grouped by reading level. For example, a recombination event can be described at a "protein" level if we are talking about domains involved in recombination, but also at a "molecular" level if we are talking about the position of the recombination on a chromosomal region. Ancestral, apomorphic, and plesiomorphic features associated with an event are

always explicitly expressed. This model is also particularly well-suited to studying automatic correlations of genetic events, and is able to correlate several events detected by DAGOBAH and temporally localized between speciation event pairs. For example, DAGOBAH may find that two genes A and B are lost twice "together" for two different lineages, which could prove very interesting in a functional perspective. In this case, if the "function" of gene A is known and the "function" of B is not, we can assume that the B gene may be involved in the "same" function as A. "By Dates" event clusters and homoplasic event clusters are the sources of a co-convergent event clustering process in DAGOBAH. For example, a "convergent evolution event cluster" is produced for events that have the same apomorphic feature objects.

The DAGOBAH ontological database must not have redundancy vs. external databases (like Ensembl; Hubbard et al. 2009). Consequently, we only model, by classes and relations, those concepts associated with specific laboratory research themes, and references were kept only to biological data or results held in external databases. The current DAGOBAH ontology adopts the Relational Ontology standard, although in the future we will probably abandon this standard so as to fully exploit the capabilities of Description Logic.

5.6 DAGOBAH Functionalities and Summarized Strategies

As described in Fig. 5.2, the strategies used in DAGOBAH can be conceptually subdivided into these different steps: (i) detection of evolutionary events, i.e., gain or loss of genes, shuffling, etc. (ii) detection of convergence between one or more gene pairs, (iii) detection of co-convergence between linked genes, (iv) search for functionally annotated gene and infer the function of correlated non-annotated gene. These four steps can be considered as forming the core of the phylogenetic comparative methods.

5.7 Detection of Events (New Architecture Appearance)

The current DAGOBAH version offers a broad panel of functions, ranging from automatic detection of genetic events to homologous domain shuffling, nonhomologous domain shuffling, insertion, deletion, gain and loss, plus gene losses and pseudogenization, and on to horizontal gene transfer and duplications (compilation on gene and species trees). A simplified summary of DAGOBAH's general strategy for event detection is:

1. Use "domain-annotated" protein alignments built from a query protein to outsource phylogeny trees building (domain trees and protein trees) to the FIGENIX platform.
2. Automatically read these trees with PhyloPattern to highlight possible events.

3. Seek to verify and clarify the putative events at a genomic level.

For new protein domain architecture events, actual examples of putative events in trees are given in the PhyloPattern publication. For this kind of event, a dedicated DAGOBAH agent studies each consecutive domain pair in the query protein architecture to investigate whether the association is the result of an event. Ideally, it finds an event's phylogenetic pattern (see Fig. 5.4) on each domain phylogenetic tree, which strengthens the event hypothesis.

The full confirmation of the event is achieved at genomic level by searching for an alignment break position between two DNA segments – one associated with the most representative apomorphic sequence and the other associated with the most representative plesiomorphic sequence. DNA segments are extracted between the domains involved (see Fig. 5.6). The most representative apomorphic sequence is chosen as the one nearest the parent node (the agent uses neighbor joining for branch lengths), while the most representative plesiomorphic sequence is chosen as the one whose domain architecture is closest to the ancestral node architecture (Dollo, Sankoff, and Mirkin parsimony algorithms (Sankoff 1975; Farris 1977; Mirkin et al. 2003) are integrated into PhyloPattern and used by the agent to infer ancestral domain architectures). If several plesiomorphic sequences share the same architecture comparison "score," the agent chooses a sequence from the nearest species in the species tree.

Gene losses and pseudogenization are studied by a set of agents in DAGOBAH, which form a module named GeneLoss. It starts the study by searching for missing species in the biggest ortholog group of the query protein tree. Each species is then studied by independent agents.

Describing the strategy in schematic terms, agents set out to determine whether the species is really missing, whether a new gene should be annotated, or whether there are some mutations or indels that can explain a pseudogenization process.

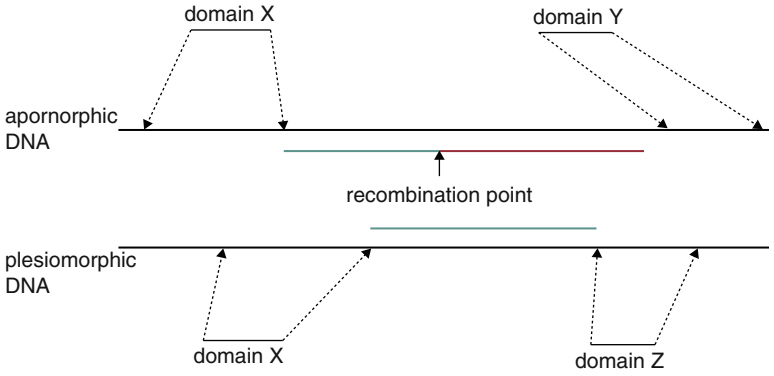

Fig. 5.6 Summary of the verification of a domain new architecture event at a genomic level. The DNA segments between domains on the apomorphic and the plesiomorphic sequences are intelligently extracted from chromosomes or scaffolds; they are then aligned and the recombination point is searched for as an alignment break

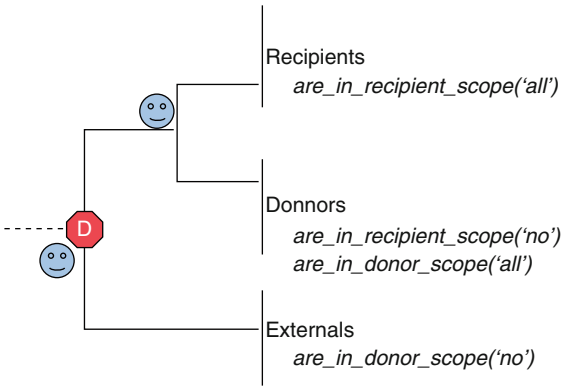

Fig. 5.7 A pattern to detect horizontal gene transfers from a phylogenetic gene tree. This means a duplication node, because the subtree does not have to match the species tree. The "donor" subtree must contain only species of a specific scope, and not from the "recipient" scope and *vice versa*

Full complex GeneLoss module strategy and results will be published separately at a later date.

Horizontal gene transfer events are detected from the query protein tree. A recipient species scope and a donor species scope are defined so as to orient the search. The dedicated agent uses PhyloPattern to annotate each internal node of the tree with two tags: are_in_recipient_scope_species and are_in_donor_scope_species, which can take three values: "no" if no species of a subtree falls in a scope, "some" if some species of a subtree fall in a scope, or "all" if all the species of a subtree fall in a scope. Then, *via* PhyloPattern, the agent applies a specific phylogenetic pattern (see Fig. 5.7) that directly gives the branch with potential HGT events.

The expert idea behind this pattern is to search the gene tree to find recipient species closer to donor species than other species that are normally placed between the recipient and donor species in the species tree.

5.8 Convergence and Co-Convergence Detection

Another important function in DAGOBAH is event convergence and co-convergence detection as conceptually described in the correlative approaches described above. Convergence identification is easy to obtain from the DAGOBAH ontological database, as a dedicated agent groups events into homoplasic convergent clusters. For example, two events are in the same convergent cluster if they have the same apomorphic character. The definition of an apomorphic character can easily be user-defined as a Prolog "ontological" pattern. The clustering mechanism is independent of the pattern definition. Co-convergence detection is a more complex task. It starts by homoplasic clustering, after which an agent produces date range clustering. Inside DAGOBAH, events are dated with tuples:

[TaxidSpeciationBefore, NumberOfDuplicationsBefore, NumberOfDuplicationsAfter, TaxidSpeciationAfter]

This tuple is determined by taking the nearest speciation event (SBE) before the event (E) on its parent branch. NumberOfDuplicationsBefore equals the number of duplication events on the branch between SBE and E. TaxidSpeciationBefore is the common parent taxid of all species in the SBE subtree. The same approach is then reapplied for the next speciation event. Date range clustering is also "user-defined" through date range patterns. Two events whose dates fit the same date pattern are pooled in the same date range cluster.

Co-convergence clusters are built with a hierarchical clustering method. A minimum co-convergent cluster is formed by four events: Eh1, Eh2, Eh1′, Eh2′. Eh1 and Eh1′ have to be in the same homoplasic cluster, while Eh2 and Eh2′ have to be in another homoplasic cluster. Eh1 and Eh2 have to be in the same date range cluster, while Eh1′ and Eh2′ have to be in another date range cluster.

We can model this basic cluster as a square:

- - - Eh1, Eh2,

- - - Eh1′, Eh2′

The clusters can be rectangular, if they come from more date clusters than homoplasic clusters (shape 1) or the opposite (shape 2). The hierarchical clustering method enables us to build the biggest possible clusters, and implies the definition of a distance method between two clusters. Our distance method favors clusters with shape 1 rather than shape 2.

Once the biggest clusters are determined, the agents seek to verify them, both statistically, *via* the Pagel method (Pagel 1994), and functionally, using the String database (Szklarczyk et al. 2011) to see whether proteins associated with events in the same homoplasic cluster belong to the same protein interactions network, and using the ArrayExpress database (Parkinson et al. 2011) to see whether proteins associated with events in the same homoplasic cluster concern the same expression experiments.

In conclusion, DAGOBAH is designed to exploit the modern functional annotation strategies and specially the evolutionary-based biology concepts. In addition, it could be addressed to various general biological questions such as searches of conserved synteny regions from a given region associated to a species to another target species.

All public results produced by DAGOBAH are openly available on the IODA Web site (http://ioda.univ-provence.fr/).

Acknowledgments This research was supported by the contract MIE (Maladies Infectieuses Emergentes-Programme Interdisciplinaire, CNRS) and ANR EvolHHuPro (ANR-07-BLAN-0054-01).

References

Aniba MR, Siguenza S, Friedrich A, Plewniak F, Poch O, Marchler-Bauer A, Thompson JD (2009) Knowledge-based expert systems and a proof-of-concept case study for multiple sequence alignment construction and analysis. Brief Bioinform 10:11–23

Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G (2000) Gene ontology: tool for the unification of biology. The gene ontology consortium. Nat Genet 25:25–29

Balandraud N, Gouret P, Danchin EG, Blanc M, Zinn D, Roudier J, Pontarotti P (2005) A rigorous method for multigenic families' functional annotation: the peptidyl arginine deiminase (PADs) proteins family example. BMC Genomics 6:153

Barker D, Pagel M (2005) Predicting functional gene links from phylogenetic-statistical analyses of whole genomes. PLoS Comput Biol 1:e3

Barker D, Meade A, Pagel M (2007) Constrained models of evolution lead to improved prediction of functional linkage from correlated gain and loss of genes. Bioinformatics 23:14–20

Collette Y, Gilles A, Pontarotti P, Olive D (2003) A co-evolution perspective of the TNFSF and TNFRSF families in the immune system. Trends Immunol 24:387–394

Danchin E, Vitiello V, Vienne A, Richard O, Gouret P, McDermott MF, Pontarotti P (2004) The major histocompatibility complex origin. Immunol Rev 198:216–232

Danchin EG, Gouret P, Pontarotti P (2006) Eleven ancestral gene families lost in mammals and vertebrates while otherwise universally conserved in animals. BMC Evol Biol 6:5

Danchin EG, Levasseur A, Rascol VL, Gouret P, Pontarotti P (2007) The use of evolutionary biology concepts for genome annotation. J Exp Zool B Mol Dev Evol 308:26–36

Eisen JA (1998) Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. Genome Res 8:163–167

Engelhardt BE, Jordan MI, Muratore KE, Brenner SE (2005) Protein molecular function prediction by Bayesian phylogenomics. PLoS Comput Biol 1:e45

Farris JS (1977) Phylogenetic analysis under Dollo's law. Syst Zool 26:77–88

Ferber J (1995) Les systèmes multi-agents. InterEdition, Paris

Force A, Lynch M, Pickett FB, Amores A, Yan YL, Postlethwait J (1999) Preservation of duplicate genes by complementary, degenerative mutations. Genetics 151:1531–1545

Gouret P, Vitiello V, Balandraud N, Gilles A, Pontarotti P, Danchin EG (2005) FIGENIX: intelligent automation of genomic annotation: expertise integration in a new software platform. BMC Bioinform 6:198

Gouret P, Thompson JD, Pontarotti P (2009) PhyloPattern: regular expressions to identify complex patterns in phylogenetic trees. BMC Bioinform 19 10:298

Haas LM, Schwarz, Kodali P, Kotlar E, Rice JE, Swope WC (2001) DiscoveryLink: A system for integrated access to life sciences data sources. IBMSJ 40:489–511.

Hubbard TJ, Aken BL, Ayling S, Ballester B, Beal K, Bragin E, Brent S, Chen Y, Clapham P, Clarke L, Coates G, Fairley S, Fitzgerald S, Fernandez-Banet J, Gordon L, Graf S, Haider S, Hammond M, Holland R, Howe K, Jenkinson A, Johnson N, Kahari A, Keefe D, Keenan S, Kinsella R, Kokocinski F, Kulesha E, Lawson D, Longden I, Megy K, Meidl P, Overduin B, Parker A, Pritchard B, Rios D, Schuster M, Slater G, Smedley D, Spooner W, Spudich G, Trevanion S, Vilella A, Vogel J, White S, Wilder S, Zadissa A, Birney E, Cunningham F, Curwen V, Durbin R, Fernandez-Suarez XM, Herrero J, Kasprzyk A, Proctor G, Smith J, Searle S, Flicek P (2009) Ensembl. Nucleic Acids Res 37:D690–D697

Levasseur A, Pontarotti P (2008) An overview of evolutionary biology concepts for functional annotation: advances and challenges. In: Pontarotti P (ed) Evolutionary biology from concept to application. Springer, Berlin, pp 209–215

Levasseur A, Pontarotti P (2011) The role of duplications in the evolution of genomes highlights the need for evolutionary-based approaches in comparative genomics. Biol Direct 6:11

Levasseur A, Gouret P, Lesage-Meessen L, Asther M, Asther M, Record E, Pontarotti P (2006) Tracking the connection between evolutionary and functional shifts using the fungal lipase/feruloyl esterase a family. BMC Evol Biol 6:92

Levasseur A, Saloheimo M, Navarro D, Andberg M, Pontarotti P, Kruus K, Record E (2010) Exploring laccase-like multicopper oxidase genes from the ascomycete trichoderma reesei: a functional, phylogenetic and evolutionary study. BMC Biochem 11:32

Mirkin BG, Fenner TI, Galperin MY, Koonin EV (2003) Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. BMC Evol Biol 3:2

Pagel M (1994) Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. Proc R Soc Lond B 255:37–45

Paillisson A, Levasseur A, Gouret P, Callebaut I, Bontoux M, Pontarotti P, Monget P (2007) Bromodomain testis-specific protein is expressed in mouse oocyte and evolves faster than its ubiquitously expressed paralogs BRD2, -3, and -4. Genomics 89:215–223

Parkinson H, Sarkans U, Kolesnikov N, Abeygunawardena N, Burdett T, Dylag M, Emam I, Farne A, Hastings E, Holloway E, Kurbatova N, Lukk M, Malone J, Mani R, Pilicheva E, Rustici G, Sharma A, Williams E, Adamusiak T, Brandizi M, Sklyar N, Brazma A (2011) ArrayExpress update–an archive of microarray and high-throughput sequencing-based functional genomics experiments. Nucleic Acids Res 39:D1002–D1004

Pellegrini M, Marcotte EM, Thompson MJ, Eisenberg D, Yeates TO (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc Natl Acad Sci USA 96:4285–4288

Rascol VL, Levasseur A, Chabrol O, Grusea S, Gouret P, Danchin EG, Pontarotti P (2009) CASSIOPE: an expert system for conserved regions searches. BMC Bioinform 10:284

Ronquist F (2004) Bayesian inference of character evolution. Trends Ecol Evol 19:475–481

Sankoff D (1975) Minimal mutation trees of sequences. SIAM J Appl Math 28:35–42

Severin J, Beal K, Vilella AJ, Fitzgerald S, Schuster M, Gordon L, Ureta-Vidal A, Flicek P, Herrero J (2010) eHive: an artificial intelligence workflow system for genomic analysis. BMC Bioinform 11:240

Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, Mungall C, Neuhaus F, Rector AL, Rosse C (2005) Relations in biomedical ontologies. Genome Biol 6:R46

Studer RA, Robinson-Rechavi M (2009) How confident can we be that orthologs are similar, but paralogs differ? Trends Genet 25:210–216

Szklarczyk D, Franceschini A, Kuhn M, Simonovic M, Roth A, Minguez P, Doerks T, Stark M, Muller J, Bork P, Jensen LJ, von Mering C (2011) The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Res 39: D561–D568

Warren DH, Pereira LM, Pereira F (1977) Prolog - the language and its implementation compared with Lisp. Proceedings of the 1977 symposium on artificial intelligence and programming languages

Wilkinson MD, Links M (2002) BioMOBY: an open source biological web services proposal. Brief Bioinform 3:331–341

Zhou Y, Wang R, Li L, Xia XF, Sun Z (2006) Inferring functional linkages between proteins from evolutionary scenarios. J Mol Biol 359:1150–1159