BioPlanner: A Plan Adaptation Approach for the Discovery of Biological Pathways across Species

Li Jin & Keith S. Decker
University of Delaware
Department of Computer and Information Sciences
Newark, DE 19716, USA
{jin, decker@cis.udel.edu}

Carl J. Schmidt
University of Delaware
Animal and Food Sciences Department
Newark, DE 19716, USA
{schmidtc@udel.edu}

Abstract

We present an implementation of a plan adaptation system, BioPlanner, built for biological pathway prediction across species. BioPlanner formulates a pathway discovery problem as a *Hierarchical Task Network* (HTN) planning problem and solves it by adapting a plan solution of another well-studied pathway. BioPlanner provides the following functionalities:

- It automatically builds HTN planning models for a biological pathway domain from the semantic web biological knowledge bases (KBs).
- It retrieves plan cases from the biological KBs.
- It generates hypothetical pathways using plan adaptation strategies with the aid of biological domain knowledge.
- It evaluates the hypothetical plan candidates, ranks them, and recommends the most likely hypotheses to users.
- It employs an information gathering multi-agent system to capture knowledge from heterogeneous sources to help the hypothetical plan generation process.

We utilize BioPlanner to predict Signaling Transduction pathways for *Mus musculus*, *Gallus gallus*, and *Drosophila melanogaster* from *Homo sapiens*.

Introduction

A biological pathway consists of the complex intercellular interactions that contribute to the function of a living cell. Because a huge amount of data about genes and gene products is generated from high throughput methods, the challenge is to place these data in the context of pathways, thus allowing biologists to make inferences about the underlying bio-processes (Barabási and Oltvai 2004). AI planning provides one approach by re-casting the task of pathway discovery as a planning problem that can be solved by planning techniques (Khan et al. 2003). Due to the hierarchical nature of the bio-processes and their underlying information, our work models a pathway with *Hierarchical Task Networks* (HTNs) (Erol, Nau, and Hendler 1994). Our approach then applies plan adaptation technologies to pathway prediction for those species with incomplete pathway information.

In this paper, we report on BioPlanner: a case-based HTN planning adaptation system to predict pathways from incomplete domain information of one species by adapting already well-known pathways of another species using plan repair strategies. BioPlanner addresses the following challenges we have to face while applying HTN planning technology to the biological pathway domain.

The first challenge we encounter is to extract the pathway domain knowledge and its subsequent representation into HTN models. Instead of the time consuming procedure of generating the task models with human experts, BioPlanner starts with existing Semantic Web data based on OWL (Dean et al. 2004), for example, the BioPAX representation for biological pathway data (Bader et al. 2005). In BioPlanner, the task decomposition formalism and plan cases are extracted automatically from Reactome (Vastrik et al. 2007), a knowledge base of manually curated *Homo sapiens* pathways.

The second challenge is incompleteness of domain knowledge that is a significant impediment of bio-pathway discovery for many species. Instead of generating pathways from raw data, BioPlanner predicts hypothetical pathway plans for those comparatively less studied species by adapting existing, curated pathways of well-studied species.

The third challenge is that many hypothetical plans might be generated for the same initial and final states but some of these hypotheses do not have real-world counterparts. BioPlanner implements an evaluation algorithm to measure the confidence of a hypothesis based on the supporting data, data resources, and the underlying adaptation or prediction methods. BioPlanner ranks the hypotheses by their confidence and recommends the best potential ones to users.

The fourth challenge is that the local domain knowledge base, created from semantic web resources, does not have complete information. Much relevant information is known to exist somewhere else. In BioPlanner, the planning engine engages a traditional multi-agent system - BioMas (Decker et al. 2002) that is responsible for gathering data from outside data resources.

This paper continues by briefly presenting the architecture of BioPlanner and then introducing the Signaling Transduction (ST) pathway domain that will be used as an example to present our approach. The next four sections focus on HTN model construction from semantic KBs, hypothetical plan generation, confidence evaluation, and multi-agent data retrieval in BioPlanner. We will utilize BioPlanner to predict ST pathways for *Mus musculus* (Mouse), *Gallus gal-*

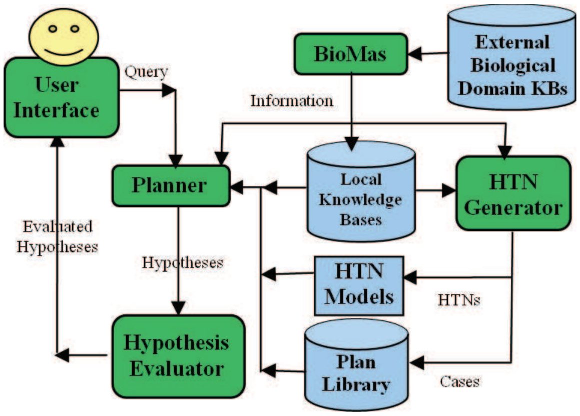


Figure 1: Information flow in BioPlanner.

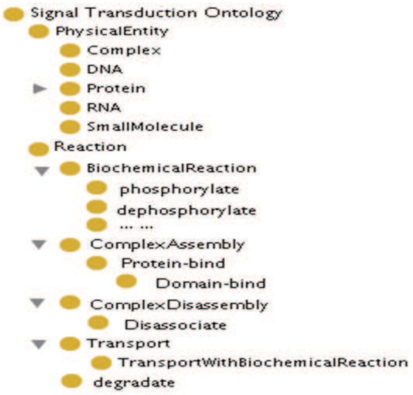

- Signal Transduction Ontology
  - PhysicalEntity
    - Complex
    - DNA
    - ▸ Protein
    - RNA
    - SmallMolecule
  - Reaction
    - ▾ BiochemicalReaction
      - phosphorylate
      - dephosphorylate
      - ... ...
    - ▾ ComplexAssembly
      - Protein-bind
        - Domain-bind
    - ▾ ComplexDisassembly
      - Disassociate
    - ▾ Transport
      - TransportWithBiochemicalReaction
    - degradate

Figure 2: Ontology of Signaling Transduction pathway.

lus (Chicken) and *Drosophila melanogaster* (Fruit fly) from *Homo sapiens* (Human). Then, we will discuss related work and finally make conclusions and discuss future work.

Overview of the System

BioPlanner is a knowledge-based planning adaptation and hypothetical evaluation system that employs plan repair strategies to attack the challenge of incomplete information in bio-pathway domains. Figure 1 shows the information flow in BioPlanner. First, the HTN generator translates the pathway knowledge stored in local KBs into HTNs and stores all the pathway plan cases into the plan library. In BioPlanner, the local pathway knowledge is from Reactome, represented in BioPAX format. The local KBs also contain other necessary data for pathway prediction, such as sequence structure data and function annotation data. When a user sends a query through the user interface, the planner retrieves the existing plan cases, adapts them, and generates hypothetical candidates. Because of incomplete domain information, the candidates might fail in the real-world environment. Therefore, a candidate should be evaluated and analyzed for potential failures. The evaluator estimates the confidence of a hypothesis based on the underlying supporting data, data resources, and repair strategies by using the evaluation algorithm that will be discussed in a later section. Finally, the user is provided with the ranked list of hypothetical solutions and information about the potential failure risks. In addition, BioMas gathers helpful information from heterogeneous external resources, e.g. DIP (Salwinski et al. 2004), to aid the planner and HTN generator.

Signal Transduction Pathway Domain

We demonstrate our planning adaptation approach with ST pathways as examples. ST pathways mostly involve cascades of protein and other molecular chemical modifications to implement information transfer across the cell. Many diseases, such as diabetes and cancers, arise from defects in ST pathways. In addition, while the malfunction of a single entity might be tolerated, the combined effect of multiple components malfunctioning can be substantial. Due to these reasons, the study of sub-cellular molecular interactions in the context of the ST pathways has vital importance to biology as well as medicine. Our discussion is relevant to other types of bio-pathways such as those of metabolism and gene reduction.

A ST pathway usually involves the following steps (Lodish et al. 2004):

1. A signaling cell secretes extracellular signaling molecules in response to external stimuli.
2. Signaling molecules transport to a target cell.
3. Signaling molecules bind and activate a specific receptor protein.
4. The activated receptor initiates a set of intracellular interactions.
5. The signal arrives at the destination and triggers functional responses, e.g. gene transcription.

Those mechanisms by which a signal is transferred through the participants within a cell can be summarized into the reactions: 1) Complex assembly: converts single bio-molecules to a complex; 2) Complex disassembly: decomposes an unstable complex into its constituent biomolecules; 3) Biochemical reaction: converts substrates to products; 4) Transport: changes the cellular location of a physical entity within a cell or between cells; and 5) Transport with biochemical reaction: changes one or more of the substrates, both their locations and their physical structures.

Figure 2 shows the ontology of the reactions and physical entities participating in a ST pathway. The physical entities participating in a ST pathway include proteins, complexes, DNAs, RNAs and small molecules. A protein can be decomposed into domain units that are associated with some functional activities and allow proteins to bind with each other forming complexes. Also, many of the intracellular portions of signaling pathways are cascades of two reactions: phosphorylation and dephosphorylation, that add or remove a phosphate group to or from a protein. One kind of protein called a catalyst can affect the confirmation of proteins by binding with them to form a transient complex, therefore activating or inhibiting the activities of those

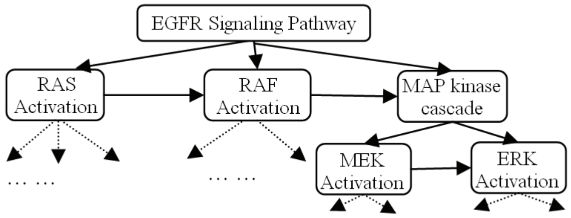


Figure 3: Hierarchical Structure of EGFR ST pathway.

proteins. Signal transduction is heavily dependent on two kinds of catalysts, kinases and phosphatases, that catalyze phosphorylations and dephosphorylations. Therefore, in our approach, we also consider some specific reactions of signal transduction including bind, phosphorylate, dephosphorylate, domain-bind, etc. The other types of signaling events that are not described here can be viewed as variations of the reactions described here.

EGFR ST Pathway In this paper, the epidermal growth factor receptor (EGFR) signalling pathway is used as an example. As shown in Figure 3, this pathway can be hierarchically decomposed into the smaller pathways: RAS activation, RAF activation, and MAP Kinase Cascade that can be further decomposed into MEK and ERK activation pathways. These smaller pathways are composed of multiple bio-reactions. The EGFR pathway is summarized as follows:

1. An epidermal growth factor (EGF) moves from plasma membrane to an extracellular region.
2. EGF binds EGFR through EGFR's extracellular domain.
3. The bound EGF and EGFR react to form a single dimer, EGF:EGFR.
4. The EGF:EGFR phosphorylates an intracellular domain at specific locations.
5. A growth factor receptor-bound protein 2 (GRB2) binds EGF:EGFR through EGFR's intracellular domain.
6. The guanine nucleotide exchange factor SOS interacts with EGFR through GRB2, and activates Ras molecule. Bio-processes 1-6 compose the RAS activation pathway.
7. Activated Ras recruits the Raf protein kinase to the membrane where Raf is phoshorylated.
8. Activated Raf binds to and phosphorylates the MEK protein, which comprise MEK activation pathway.
9. Phosphorylated MEK binds the ERK protein and phosphorates it. Activated ERK transports to the nucleus, where it triggers new gene expression. These bio-processes occur through the reactions of ERK activation pathway.

Constructing HTN Planning Models for ST Pathways

Larger pathways can be built by combining individual parameterized modules of individual ST pathways (Endy and Brent 2001). HTN provides one solution, where tasks representing independent pathways are combined within a network to form a larger pathway. Khan et al. have shown that hierarchical representations can help with scalability compared to STRIPS planning (Khan et al. 2003). However, they have noticed that hand-coding pathway knowledge into HTN task models is time consuming. In this paper, we have developed an HTN generator that can construct ST pathway HTN task models from Reactome.

We follow the principles of HTNs using the SHOP2 system (Nau et al. 2005) to perform hierarchical decompositions of a pathway. HTN planning achieves complex tasks by decomposing them into simpler subtasks. Planning continues by decomposing the simpler tasks recursively until tasks representing concrete actions are generated. These actions form a plan achieving the high-level task specifications.

Generally, an ST pathway can be recast as an HTN planning problem (I, T, D), where:

- I is an initial state that is the conjunction of the initial configurations of the pathway components. For example, each protein is initialized to some state, such as its cellular location.
- T is a task to transfer information from one location to another initialized by a specific kind of protein. For example, the EGFR pathway can be considered as a task to transfer information initialized by EGF between cells.
- D is the domain theory that is a collection of operators and methods.

A plan solution is a sequence of biological reactions whose executions are the biological processes responding to stimulus events. The information of a ST pathway stored in Reactome, such as the EGFR pathway, can be mapped to the HTNs formalism as the following.

Physical Entities

Any individual physical entity (*PE*) participating in a pathway is automatically mapped to a predicate *(istype PE t)* indicating that *PE* is of type *t*, where *t* stands for one of the physical entity types discussed in the previous section. A variable *?x* of a type *t* is defined using a predicate *(istype ?x t)*. A protein can be further decomposed in terms of domains. The relationship between a protein and its composing domain units can be described as *(has-domain p d)* indicating *p* has a domain, *d*. For example, *EGFR* is a protein with three domains. This is specified by *(istype EGFR protein)*, *(istype EGFR-extra domain)*, *(has-domain EGFR EGFR-extra)*, etc. The ontology of physical entities described in BioPAX can also be represented by a predicate *(isa subtype type)*. For instance, *(isa protein physical-entity)* indicates that protein is a subtype of physical entity.

Compartmentalization

Compartmentalization describes a specific location of a cell where a physical entity would function. The cellular location information of an entity is mapped to a predicate, *(in physical-entity location)*. For example, *EGF* is present at the plasma membrane, thus its location is described by *(in EGF plasma-membrane)*.

Hierarchical Abstract Operators

In HTNs, an operator *O* is of the form *(h, pre, dl, al)*, such that: *h* is the operator's head, *pre* is a set of preconditions, and *dl* (delete-list) and *al* (add-list) are effect lists that define how the operator transforms the current state. In our approach, the general biological reactions that modify the states of physical entities in a cell are mapped to abstract operators. Each abstract operator has the name of a reaction as its head and a list of variables representing participating physical entities and their cellular locations as parameters. In addition, the states of the inputs and those of the outputs of a reaction are added into the operator's *dl* and *al*, respectively. The precondition set of an operator contains the types and the states of input entities. The operators are managed in a hierarchical structure based on the differences in the biochemical detail they reflect. For example, domain-bind provides a more detailed characterization of proteins and models the reaction at the level of the protein domain units.

For instance, the protein-bind reaction can be explicitly modeled by the following protein-bind operator. It requires two proteins in the same cellular location, where *(can-ppi ?x ?y ?loc)* allows only those reactions that exists in PPI (Protein-Protein Interaction) KBs:

```
(:operator (!protein-bind ?x ?y ?loc)
 (;;precondition
    (istype ?x protein)
    (istype ?y protein)
    (istype ?loc compartment)
    (in ?x ?l)
    (in ?y ?l)
    (can-ppi ?x ?y ?loc))
 (;;delete-list
    (in ?x ?loc)
    (in ?y ?loc))
 (;;add-list
    (istype ?x:?y protein-bound)
    (in ?x:?y ?l)))
```

Method Model

The method model is the union of decomposition descriptions that define how to decompose a complex pathway and how to complete a biological process. A method, *M*, is a 3-tuple: *(h, P, SubT)*, such that: *h*, called the head of *M*, is the task being decomposed; *P* is the precondition set required for using the method; and *SubT* is the set of the subtasks achieving *h*. A Reactome pathway is represented as a hierarchical task decomposition model in our approach. A pathway's name is mapped to the head of a method and its hierarchical components are mapped to its subtasks. The precondition set contains those preconditions that cannot be achieved by any other subtasks, including those catalysts that activate/inactivate the reactions and those physical entities that are not output from any reactions in the pathway.

The plan cases and action cases of ST processes that can take place in a real cell are extracted by the HTN generator from Reactome and other knowledge bases (e.g. (Salwinski et al. 2004)). These cases are stored in the plan library and will be modified to solve new problems.

Generating Hypothetical Plans

For many species we do not have enough information available for pathway construction from scratch. Instead, BioPlanner adapts the well-studied pathways of humans to predict similar pathways for other species by using plan repair strategies from previous works (e.g. (Hammond 1990; Kambhampati and Hendler 1992)).

To solve a problem by adaptation, the first step is to search for suitable candidates based on the similarity of a user's query and the plan cases in the library. For this purpose, when a plan case is stored in the plan library, its summarized annotation information is also saved with it, including its name, participants, initial states, final states, external preconditions that cannot be achieved by the plan's subtasks, and the species it belongs to. When a plan candidate case (called a *reference plan*) is found, BioPlanner will adapt it using the following strategies.

Action Modification

Generally, when a reference pathway plan is applied to another species (called a *target species*), most of its actions have to be modified because many of the participating physical entities of the original actions do not exist in the target species, causing the actions to fail.

Strategy 1: Physical entity adaptation BioPlanner modifies the failing actions by replacing the participating physical entities of the reference plan with those entities that have similar physical structures and can be responsible for the actions in the target species. Sequence alignment methods (e.g. BLAST(Altschul et al. 1990)) are employed for this strategy. Suitable substitutes having similar biochemical functions and similar sequence structures will be identified from the knowledge bases of the target species. A list of substitute candidates is generated with a confidence evaluation (e-value) that is provided by the alignment method. The e-value can be used to measure the confidence of a hypothetical action adapted by Strategy 1. This confidence can help users estimate the possibility that the hypothesis will happen in the real environment of the target species.

Strategy 2: Replacing an action If a failing action can not be repaired by Strategy 1, the searching process will continue to search for a replacing action in the plan library that can accomplish the primitive task. The new action also has a confidence value depending on the supporting data and data resources. As an example, for the action *(react P1 P2 P3)* indicating that *P1* reacts with *P2* producing *P3*, if Strategy 1 can adapt *P1* and *P3* to *P1'* and *P3'* without finding any substitute protein for *P2*, but Strategy 2 can find *(react P4' P5' P3')* that can produce *P3'*, then the original action will be adapted by *(react P4' P5' P3')*.

Task Modification

If a plan candidate still has failing actions that can not be repaired by action adaptation strategies, the adaptation procedure will go into the task-level.

Strategy 3: Splitting an action In a biological process, the expectations not achieved by one reaction might be achieved by multiple reactions. Therefore, one possible way to repair a failing action is splitting this action into multiple actions that can achieve the primitive task. We can consider this Strategy as recasting the primitive task as a compound task and re-decomposing it into multiple primitive tasks to achieve the expected effects. For instance, a failing action *(protein-bind P1 P2)* can be adapted by *(protein-bind P1 P3)* and *(protein-bind P3 P2)* so that *P1* can bind with *P2* through *P3*.

Strategy 4: Combining actions On the other hand, the results produced by multiple reactions can be generated by one reaction. Thus, this strategy considers re-decomposing a task by combining its several primitive tasks into one primitive task that can be accomplished by one action.

Strategy 5: Adding a new task A method *M* of a task may contain some specifications of preconditions that will not be achieved in any way by continually decomposing the method *M* into a sub-plan, but must be satisfied before the method is applied. For example, a catalyst of a plan must be satisfied before the subtask or action that requires the catalyst begins. For these preconditions that are not satisfied in the adapted plans, one possible modification is to find a task that can establish them before the tasks that require the preconditions. If a task cannot generate the expected effects, a new subtask might be added to achieve the failing effects.

Strategy 6: Redecomposing a task Another method to repair a failing task is to find an alternative decomposition method that does not require the failing preconditions or can achieve the failing effects.

Ordering Modification

Strategy 7: Adapting orders This repair strategy concerns the ordering of biological reactions. For example, one reaction may damage some effects of some previous reactions, or one reaction may destroy the preconditions of later reactions. Therefore, one potential repair method is to change the orderings of the actions or tasks in a plan.

During the adaptation procedure, each action of a hypothetical plan has the adaptation method and the related information stored with it. The information will be used to evaluate the hypothetical plan and to explain to a user how the plan repair proceeded.

Evaluating Hypotheses

Many hypotheses might be generated for one task while some of the candidates might not have real-world counterparts. Therefore a hypothetical plan has to be ascribed a confidence measure. Because it is difficult to quantitatively compare the confidences of different methods, we have developed a ranking algorithm to list the candidates in order

```
function SORT($HPS$)
input  : a set of hypothetical plans $HPS$
output : Sorted $HPS$ based on its elements' confidence
$S \longleftarrow \emptyset$
foreach hypothetical plan $h \in HPS$ do
    foreach element $e \in S$ do
        if ConfidenceCompare($h$, $e$)>0 then
            insert $h$ into $S$ before $e$
        else
            if $e$ is the last element in $S$ then
                insert $h$ into $S$ after $e$
    endfor
endfor
return $S$

function ConfidenceCompare($hp1$, $hp2$)
input  : hypothetical plan $hp1$, $hp2$
$A1 \leftarrow$ action set of $hp1$
$A2 \leftarrow$ action set of $hp2$
$NF1 \leftarrow$the number of failing actions in $A1$
$NF2 \leftarrow$the number of failing actions in $A2$
if $NF1$/sizeof($A1$)<$NF2$/sizeof($A2$) then
    return 1
else if $NF1$/sizeof($A1$)>$NF2$/sizeof($A2$) then
    return -1
if RelyVal($hp1$, $A1$)>RelyVal($hp2$, $A2$) then
    return 1
else if RelyVal($hp1$, $A1$)<RelyVal($hp2$, $A1$) then
    return -1
for $i \leftarrow$4 to 2 do
    if Strategyi($hp1$) <Strategyi($hp2$) then return 1
    if Strategyi($hp1$)>Strategyi($hp2$) then return -1
endfor
if AvgEVal($hp1$)<AvgEVal($hp2$) then return 1
else if AvgEVal($hp1$)>AvgEVal($hp2$) then return -1
return 0

function RelyVal($hp$, $A$)
input  : hypothetical plan $hp$ and its action set $A$
$value \leftarrow 0.0$
foreach element $a \in A$ do
    if the biological reaction corresponding to a can be found
    in knowledge bases then
        $value = value + 1.0$
endfor
return $value/sizeof(A)$

function AvgEVal($hp$, $A$)
$Sum \leftarrow 0.0$
$Np \leftarrow 0$
foreach element $a \in A$ do
    foreach participant $p$ participating in $a$ do
        $Sum = Sum$+ e-value of $p$
        $Np++$
    endfor
endfor
return $Sum/Np$

function Strategyi($hp$, $A$)
$N \leftarrow$ the number of actions in $A$ adapted by Strategy $i$
return $N/sizeof(A)$
```

Algorithm 1: Hypothetical Plan Ranking Algorithm

of their comparative confidence estimated based on the supporting data, resources and the modifications taken place during the adaptation process.

We have developed a confidence evaluation algorithm based on the biological assumption that a hypothetical pathway is more preferred if it has fewer differences from the original pathway. The difference takes into account the participants' structures and functions, the reactions, and the pathway decomposition. In the bio-domain, a hypothetical reaction found in literature or experimental resources is considered more confident. In addition, a hypothetical pathway obtained only by physical entity substitutions is thought more confident than those containing other modifications. If a hypothesis is achieved by splitting a failing reaction into two reliable reactions, it is more confident than splitting the failing reaction into more than two reactions.

Based on theses biological assumptions, we assign the priorities of the adaptation strategies in the decedent order, Strategy 1 > Strategy 2 > Strategy 3 = Strategy 4 > Strategy 5 = Strategy 6. We have developed hypothesis ranking algorithm as shown in Algorithm 1, where two hypothetical pathways, *hp1* and *hp2*, can be compared for their confidence by the following rules:

- A hypothetical plan might contain failing actions that cannot be repaired. If *hp1* has a lower percentage of failing actions than *hp2*, then *hp1* is ranked more confident than *hp2*.
- *hp1* is assumed more confident than *hp2* if *hp1* has a higher percentage of actions whose corresponding reactions can be found in protein-protein interaction KBs or literature resources.
- *hp1* is assumed more confident if *hp1* contains a higher percentage of actions that are achieved by applying adaptation strategies having higher priorities than those strategies used in *hp2*.
- If *hp1* and *hp2* contain the same percentage of actions adapted from the same strategies, then *hp1* is thought more confident if *hp1* has a lower average e-value of participating entites than *hp2* or if *hp1* has more actions adapted by using more reliable resources. The e-value comes from the underlying sequence comparison method. The resources BioPlanner relies on are ranked by their reliability.

Multi-Agent System

BioPlanner can only generate new and promising hypotheses if it is applied to new data that can improve and/or modify existing plans. The volume of data collected requires that the acquisition and translation of data into the planning formalism be automated. For these reasons, we have incorporated the planning engine into a multi-agent system that gathers data from multiple sources and populating the local knowledge base for the planner and the generator.

Figure 4 shows the integration of BioMas that is responsible for data gathering and other components in BioPlanner. The manager agent is responsible for finding some agent to gather more information for the planner. Agent name servers, matchmakers, and other domain independent agents support the creation of open systems where elements may come and go over time. The planner agent and HTN Model Generator agent wrap the planning and generating components in BioPlanner. The local KB Management Agent is responsible for interacting with local knowledge bases for storing and retrieving data.

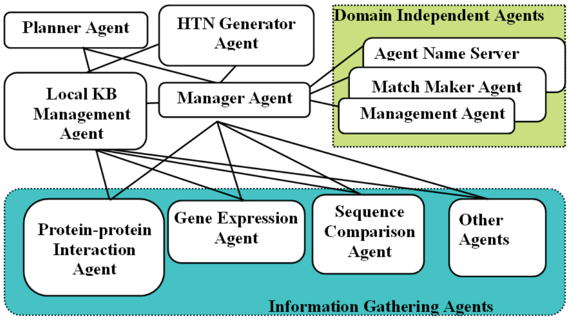

Planner Agent
HTN Generator Agent
Domain Independent Agents
Agent Name Server
Match Maker Agent
Management Agent
Local KB Management Agent
Manager Agent
Protein-protein Interaction Agent
Gene Expression Agent
Sequence Comparison Agent
Other Agents
Information Gathering Agents

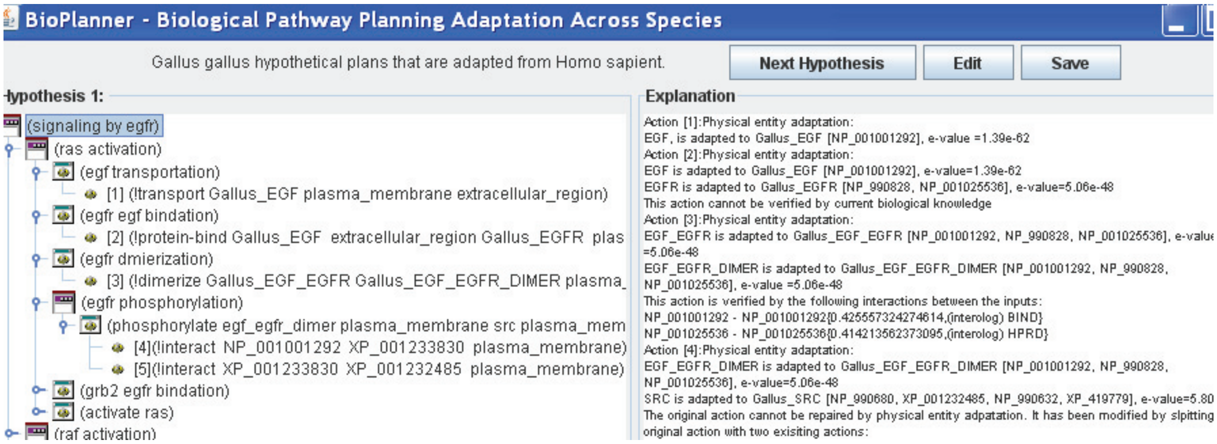
Figure 4: Multi-agent data gathering system in BioPlanner.

Implementation and Experiments

BioPlanner is implemented on JSHOP2 (Nau et al. 2005) with the additional components of adaptation, evaluation, etc. BioPlanner has integrated data gathered from 11 knowledge resources. The ST pathway HTN model currently consists of 14 operator schemas. Around 470 action cases and 150 plan cases of Human pathways have been retrieved from Reactome and stored in the plan library. Based on these cases, we have applied BioPlanner to predict hypothetical pathways for three species: Mouse, Chicken and Fruit fly.

Figure 5 is a snapshot of the hypothetical pathway generated by BioPlanner for Chicken by adapting the Human EGFR pathway. In the figure, the left side shows the hierarchical structure of the plan, the right side illustrates the information related to the plan adaptation, such as the information about the method used for action adaptation, the resources where the supporting data came from, and the reliability of the supporting data or resources. Even when plan generation fails, the partially repaired hypotheses will show up with messages telling a user the components of the original plan that cannot be repaired and the reasons. In summary, BioPlanner tries its best to present to a user as much information as possible that might be helpful.

Figure 6 shows the percentage of pathways that can be completed using a combining similar sequence prediction with hypothesis generation. As expected the percentage of completed pathways decreases with evolutionary distance from the human. Unfortunately, there are no completely curated pathway database to allow evaluation of these hypotheses. Current effort focuses on using the hypotheses to design laboratory experiments thus allowing direct evaluation of these predictions. Figure 7 shows the performance of BioPlanner corresponding to scalability. As the pathway size measured in terms of the number of the component reactions increases, the average running time needed to find a


BioPlanner - Biological Pathway Planning Adaptation Across Species
Next Hypothesis
Edit
Save

Gallus gallus hypothetical plans that are adapted from Homo sapient.

Hypothesis 1:

```
(signaling by egfr)
  (ras activation)
    (egf transportation)
      [1] (!transport Gallus_EGF plasma_membrane extracellular_region)
    (egfr egf bindation)
      [2] (!protein-bind Gallus_EGF  extracellular_region Gallus_EGFR  plas
    (egfr dmierization)
      [3] (!dimerize Gallus_EGF_EGFR Gallus_EGF_EGFR_DIMER plasma_
    (egfr phosphorylation)
      (phosphorylate egf_egfr_dimer plasma_membrane src plasma_mem
        [4](!interact  NP_001001292  XP_001233830  plasma_membrane)
        [5](!interact  XP_001233830  XP_001232485  plasma_membrane)
    (grb2 egfr bindation)
    (activate ras)
  (raf activation)
```

Explanation

```
Action [1]:Physical entity adaptation:
EGF, is adapted to Gallus_EGF [NP_001001292], e-value =1.39e-62
Action [2]:Physical entity adaptation:
EGF is adapted to Gallus_EGF [NP_001001292], e-value=1.39e-62
EGFR is adapted to Gallus_EGFR [NP_990828, NP_001025536], e-value=5.06e-48
This action cannot be verified by current biological knowledge
Action [3]:Physical entity adaptation:
EGF_EGFR is adapted to Gallus_EGF_EGFR [NP_001001292, NP_990828, NP_001025536], e-value
=5.06e-48
EGF_EGFR_DIMER is adapted to Gallus_EGF_EGFR_DIMER [NP_001001292, NP_990828,
NP_001025536], e-value =5.06e-48
This action is verified by the following interactions between the inputs:
NP_001001292 - NP_001001292{0.425557324274614,(interolog) BIND}
NP_001025536 - NP_001025536{0.414213562373095,(interolog) HPRD}
Action [4]:Physical entity adaptation:
EGF_EGFR_DIMER is adapted to Gallus_EGF_EGFR_DIMER [NP_001001292, NP_990828,
NP_001025536], e-value=5.06e-48
SRC is adapted to Gallus_SRC [NP_990680, XP_001232485, NP_990632, XP_419779], e-value=5.80
The original action cannot be repaired by physical entity adpatation. It has been modified by slpitting
original action with two exisiting actions:
```

Figure 5: Snapshot of a hypothetical EGFR pathway generated by BioPlanner for *Gallus gallus*.

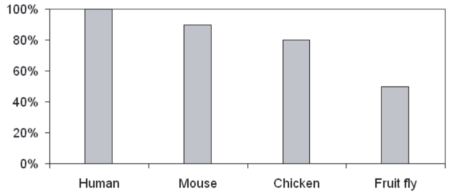


Figure 6: % of Human ST pathways repaired for other species.

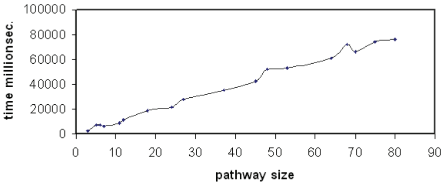


Figure 7: Running time to generate hypothetical pathways of different sizes.

solution also increases.

Related Work

Applying AI planning technology to biological pathway discovery is first presented in (Khan et al. 2003) where a deterministic classical planner is used to solve the problem. Along this planning trend, (Tran and Baral 2005) show that the changes in cellular processes can be modeled as exogenous actions called triggers. (Bryce and Kim 2007) present a formulation of Gene Regulatory Network intervention as a decision theoretical planning problem. Different from these works, BioPlanner does not plan from scratch, but uses replanning techniques and generates hypothetical plans with incomplete domain models. Consequently, it does not require a full description of the signal pathway domain in question. Therefore, BioPlanner can help biologists discover new pathways across species from the existing pathways, with the help of computational analysis.

Many researchers have shown the advantages of plan adaptation (Nebel and Koehler 1995), and plan adaptation and repair techniques have been developed and applied to many real-world domains (e.g. (Hanks and Weld 1995; Wilkins 1985; Beetz and McDermott 1994; Warfield et al. 2007)). Similar to those previous case-based adaptation works (e.g. (Veloso and Carbonell 1993; Muñoz-Avila and Cox 2008)), our approach also modifies the previous cases to solve new problems based on domain knowledge. In addition to the adaptation strategies summarized from these works, BioPlanner has integrated information gathering and confidence evaluation in order to attack the challenges of an incomplete biological information domain.

In the HTN planning field, various approaches have been applied to construct HTN models, such as learning task models from STRIPS cases (Hogg, Muñoz-Avila, and Kuter 2008) or from cases with the hierarchical relationships between tasks already known (Muñoz-Avila et al. 2001; Ilghami et al. 2005). HTNs can also be extracted from OWL-based information (Sirin et al. 2004). Our approach takes advantage of existing OWL representations of pathway knowledge to generate HTN models from pathway cases.

Authors have observed that plan evaluation can play important role in plan repair (Fox et al. 2006; Garland and Lesh 2002). Fox et al. process the evaluation based on the differences of the actions between the original plan and the new one. Garland and Lesh increase the confidence of a plan by minimizing the actions of incomplete information. Our approach evaluates a hypothesis based on its supporting data and resources. Instead of only considering the actions, BioPlanner also takes task method evaluation into account.

There exist some other approaches for bio-pathway study, e.g. EcoCyc (Karp 2001), Petri Nets (Peleg, Yel, and Altman 2002) and Statecharts (Harel and Gery 1997). However, these works require the fully described state and state translation descriptions that are not satisfied by the bio-domains of many species.

Conclusion and Future Work

In this paper, we have described several challenges while applying an AI planning approach to biological pathway domain, and we have showed how BioPlanner meets these challenges. In our knowledge, BioPlanner is the first plan adaptation and evaluation system that generates hypothetical pathways across species. These hypotheses provide biologists with useful information about uncurated biological processes. We will evaluate our approach further when more curated pathway data is available. We also intend to develop a diagnosis component for a hypothetical plan execution in order to help users analyze the differences between real-world experimental data and expected results from a hypothetical simulation.

Acknowledgment

This work is supported in part by the Cooperative State Research, Education, and Extension Service, U.S. Department of Agriculture, under Agreement No. 2008-35205-18734.

References

Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; and Lipman, D. J. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215(3):403–410.

Bader, G.; Cary, M.; Aladjem, M.; and et al. 2005. Biopax: Biological pathways exchange language level 2, version 1.0. http://www.biopax.org/.

Barabási, A.-L., and Oltvai, Z. 2004. Network biology: Understanding the cellss functional organization. *Nature Reviews Genetics* 5:101–113.

Beetz, M., and McDermott, D. 1994. Improving robot plans during their execution. In *Proceedings of AIPS'94*, 3–12.

Bryce, D., and Kim, S. 2007. Planning for gene regulatory network intervention. In *Proceedings of IJCAI'07*.

Dean, M.; Schreiber, G.; McGuinness, D. L.; van Harmelen, F.; and et al. 2004. Owl web ontology language reference. http://www.w3.org/tr/2004/rec-owl-ref-20040210/.

Decker, K.; Khan, S.; Schmidt, C.; and et al. 2002. Biomas: A multi-agent system for genomic annotation. *Intl. J. of Coop. Info. Sys.* 11:265–292.

Endy, D., and Brent, R. 2001. Modelling cellular behaviour. *Nature* 409:391–395.

Erol, K.; Nau, D.; and Hendler, J. 1994. HTN planning: Complexity and expressivity. In *Proceedings of AAAI'94*, 123–128.

Fox, M.; Gerevini, A.; Long, D.; and Serina, I. 2006. Plan stability: Replanning versus plan repair. In *Proceedings of ICAPS'06*, 212–221.

Garland, A., and Lesh, N. 2002. Plan evaluation with incomplete action descriptions. In *Proceedings of AAAI'02*.

Hammond, K. 1990. Explaining and repairing plans that fail. *Artificial Intelligence* 45:173–228.

Hanks, S., and Weld, D. 1995. A domain-independent algorithm for plan adaptation. *Journal of Artificial Intelligence Research (JAIR)* 2:319–360.

Harel, D., and Gery, E. 1997. Executable object modeling with statecharts. *IEEE Computer* 30:31–42.

Hogg, C.; Muñoz-Avila, H.; and Kuter, U. 2008. HTN-MAKER: Learning HTNs with minimal additional knowledge engineering required. In *Proceedings of AAAI'08*.

Ilghami, O.; Muñoz-Avila, H.; Nau, D.; and Aha, D. 2005. Learning approximate preconditions for methods in hierarchical plans. In *Proceedings of ICML'05*.

Kambhampati, S., and Hendler, J. 1992. A validation structure-based theory of plan modification and reuse. *Artificial Intelligence* 55:193–258.

Karp, P. 2001. Pathway databases: A case study in computational symbolic theories. *Science* 293(5537):2040–2044.

Khan, S.; Gillis, W.; Schmidt, C.; and Decker, K. 2003. A multi-agent system-driven AI planning approach to biological pathway discovery. In *Proceedings of ICAPS'03*.

Lodish, H.; Berk, A.; Zipursky, S.; and et al. 2004. *Molecular Cell Biology*. New York: W.H.Freeman and Company.

Muñoz-Avila, H., and Cox, M. 2008. Case-based plan adaptation: An analysis and review. *IEEE Intelligent Systems* 23(4):75–81.

Muñoz-Avila, H.; Aha, D. W.; Nau, D. S.; Breslow, L. A.; Weber, R.; and Yamal, F. 2001. SiN: Integrating case-based reasoning with task decomposition. In *Proceedings of IJCAI'01*. AAAI Press.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Muñoz-Avila, H.; Murdock, J.; Wu, D.; and Yaman, F. 2005. Applications of SHOP and SHOP2. *IEEE Intelligent Systems* 20(2):34–41.

Nebel, B., and Koehler, J. 1995. Plan reuse versus plan generation: A theoretical and empirical analysis. *Artificial Intelligence* 76(1-2):427–454.

Peleg, M.; Yel, I.; and Altman, R. 2002. Modeling biological processes using workflow and petri netmodels. *Bioinfo. J.* 18(6):825–837.

Salwinski, L.; Miller, C.; Smith, A.; Pettit, F.; Bowie, J.; and Eisenberg, D. 2004. The database of interacting proteins. *Nucleic Acids Research* 32:449–451.

Sirin, E.; Parsia, B.; Wu, D.; Hendler, J.; and Nau, D. 2004. HTN planning for web service composition using SHOP2. *Journal of Web Semantics* 1(4):377–396.

Tran, N., and Baral, C. 2005. Issues in reasoning about interaction networks in cells: necessity of event ordering knowledge. In *Proceedings of AAAI'05*.

Vastrik, I.; D'Eustachio, P.; Schmidt, E.; Joshi-Tope, G.; and et al. 2007. Reactome: A knowledge base of biologic pathways and processes. *Genome Biology* 8.

Veloso, M., and Carbonell, J. 1993. Planning and learning by analogical reasoning. *Machine Learning* 10:249–278.

Warfield, I.; Hogg, C.; Lee-Urban, S.; and Munoz-Avila, H. 2007. Adaptation of hierarchical task network plans. In *Proceedings of FLAIRs'07*. AAAI Press.

Wilkins, D. 1985. Recovering from execution errors in SIPE. *Computational Intelligence* 1:33–45.