2011 IEEE International Conference on Services Computing

ServiceMap: Providing Map and GPS Assistance to Service Composition in Bioinformatics

Wei Tan[1], Jia Zhang[2], Ravi Madduri[3], Ian Foster[3], David De Roure[4], Carole Goble[5]
[1] *IBM T.J. Watson Research Center, USA*
[2] *Department of Computer Science, Northern Illinois University, USA*
[3] *Mathematics and Computer Science Division, Argonne National Laboratory, USA*
[4] *Oxford e-Research Centre, Oxford University, UK*
[5] *School of Computer Science, University of Manchester, UK*
wtan@us.ibm.com, jiazhang@cs.niu.edu, madduri@mcs.anl.gov, foster@mcs.anl.gov, david.deroure@oerc.ox.ac.uk, carole.goble@manchester.ac.uk

Abstract**—***The wide use of Web services and scientific workflows has enabled bioinformaticians to reuse experimental resources and streamline data processing. This paper presents a follow-up work of our network analysis on the myExperiment, an online scientific workflow repository. The motivation comes from two common questions raised by bio-scientists: 1) Given the services that I plan to use, are there other services usually used together with them? and 2) Given two or more services I plan to use together, is there an operation chain to connect them based on others' past usage? Aiming to provide a system-level GPS-like support to answer the two questions, we present ServiceMap, a network model established to study the best practice of service use. Two approaches are proposed over the ServiceMap: association rule mining and relation-aware, cross-workflow searching. Both approaches were validated using the real-life data obtained from the myExperiment repository.*

I. INTRODUCTION

In many disciplines including biomedicine and bioinformatics, Web services have been widely used as virtualized access points to various data and computational resources [1]. To accelerate data-intensive scientific exploration, services are orchestrated into scientific workflows that precisely describe a composition of tasks and the dataflow among them [2]. Over the past three years we participated in the development of Cancer Biomedical Informatics Grid (caBIG) [3], a platform for cancer researchers to share service-based capabilities. We have extended the Taverna Workbench [4] into the *caGrid Workflow Toolkit* [5], a software suite for bioinformaticians to compose and coordinate caBIG services as well as third-party Web services.

Given the caGrid Workflow Toolkit and other tools in the biomedical workflow domain, however, bioinformaticians are still hesitated to exploit the existing services. One major reason is that they are unaware of the usage patterns of available services [6], thus cannot effectively incorporate the best practices when they build new workflows. This paper presents a follow-up work of our network analysis on the myExperiment [7], an online biological workflow repository.

Our idea is inspired by the Global Positioning System (GPS) that is capable of recommending travel paths between locations. An idea of ServiceMap is introduced, where services are modeled as locations in a map and their past usage connections (in common workflows) are modeled as transportation paths between them. Based on the ServiceMap, we aim to provide a GPS-like support to: 1) help domain scientists better understand various usage patterns of the existing services; and 2) provide a system-level support to recommend possible service compositions. The former objective shall help domain scientists gain confidence of using available services; and the latter will pave a new pathway toward automatic service composition. Note that although our project focuses on studying biomedical services and workflows, our approaches can be applied to other domains.

The remainder of the paper is organized as follows. Section II discusses the motivation of our research. Section III describes the overall ServiceMap approach. Section IV presents an association mining-based method for suggesting relevant services. Section V presents a relation-aware, cross-workflow searching method for suggesting operation chains. Section VI presents an empirical study. Section VII discusses related work and Section VIII draws conclusions.

II. RESEARCH MOTIVATION

Our experience in caBIG and other service-based bioinformatics grids shows that, although the sharing of service-based capabilities opens a gateway to resource reuse, the mission is still far from accomplished. We have conducted a study [6] over the workflows stored in the myExperiment, a public scientific workflow repository. Applying social network analysis techniques, we examined the interaction patterns between the workflows and comprising services. This state-of-the-art analysis revealed that biomedical services are currently used in an *ad hoc* style and with low reusability. This suggests a need for techniques that help domain scientists better locate interested services and workflow fragments and reuse them to attain their research purposes.

Our experience in caBIG also summarizes two questions that domain scientists frequently ask when they try to exploit external Web services in building a scientific workflow:

Q1: *Given the services I plan to use, what are the other services often used together with them, by other scientists?*

Q2: *Given two or more services I want to use together, can I find an operation chain, which is already used by others, to connect them?*

Q1 is usually raised when a scientist first reaches some data or analytical capabilities wrapped and exposed as services. Because he is new to the services, the scientist intends to know how his peer scientists use them together with other services of which he may or may not be aware. Besides, due to the explorative nature of scientific workflows, incorporating a newly found service with known ones may lead to new research ideas. For example, assume that a scientist plans to leverage a statistical analysis service over microarray data to find significantly expressed genes in some tissue. However, he is unaware of a gene pathway service that is able to expose interactions among genes. If he can be prompted that his peer scientists frequently use the pathway service together with the statistical one, he knows that from the microarray data he has at hand, he can identify not only significant genes but also their interaction patterns.

Q2 is usually asked when a scientist's experiment procedure becomes complex and requires to use multiple external services. In the aforementioned microarray experiment, a scientist intends to exploit three services: one for generating microarray data; one for analyzing microarray data; and one for retrieving pathway information. However, linking the three services is not trivial, e.g., microarray data have to be cleaned before being input to the statistical service; access to the pathway service requires a special security mechanism; and so on. The scientist thus wishes to know the exact sequence in which these services can be chained together.

In real-life transportation systems, two questions Q1' and Q2' are frequently asked against a map or a GPS system.

Q1': *Do people who visit some places also visit others?*

Q2': Can I *find a route between two places?*

Q1' and Q2' become counterparts of the questions Q1 and Q2, respectively, if we make the following analogies: 1) service operations vs. places in a map, and 2) scientific workflows (or service compositions) vs. streets/routes in a map. Consequently, the questions asked by scientists resemble those that can be solved by a map or a GPS system (i.e., to recommend places to visit or find a route two places). This analogy inspires us to take a step further from our previous network analysis of the myExperiment, by building a service network and associated facility to address Q1 and Q2, just like how a map/GPS system addresses Q1' and Q2'.

III. SERVICEMAP APPROACH

As depicted in the previous example, Q1 and Q2 are not isolated; instead, they may be raised in different stages of an *in-silico* experiment. Q1 is usually raised when an experiment is in its conceptual design level and scientists need to figure out the available resources to leverage. Q2 is raised in a later stage when the routine of an experiment needs to be concretized in the operational level. It is also a common practice to iterate over these two stages due to the explorative nature of bioinformatics experiments. Inspired by this requirement from multiple user communities including caBIG, we propose ServiceMap as a framework to address both Q1 and Q2, in a holistic manner. Figure 1 illustrates the general approach of ServiceMap. We first download all the myExperiment workflows. We abstract these workflows by removing non-Web (i.e., WSDL-based) services, such as local beanshells, xml manipulating blocks, while maintaining the data flows between services. Afterwards these abstract workflows are combined into ServiceMap. ServiceMap consists of two disjoint networks (graphs): an undirected *workflow-service network* and a directed *operation network.*

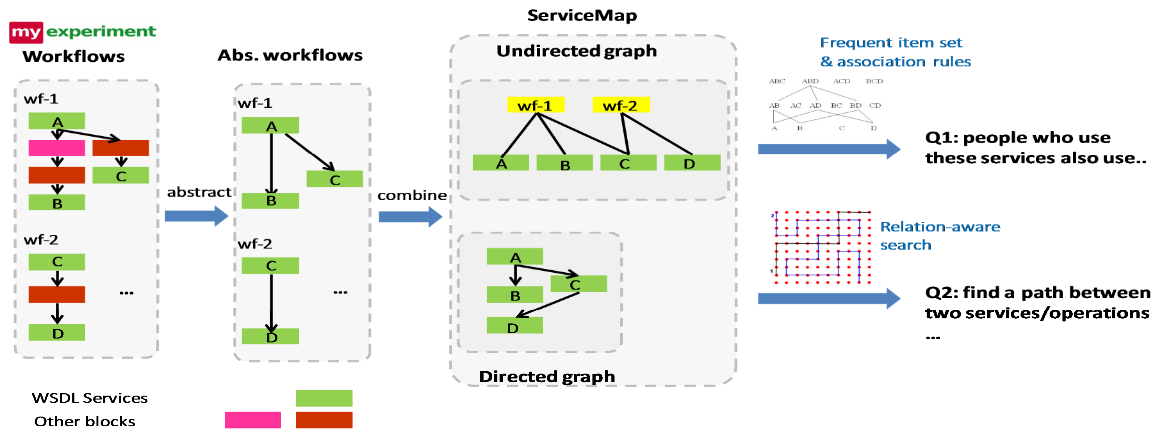


Figure 1. ServiceMap approach to answer Q1 and Q2.

In the undirected workflow-service network, nodes are either workflows or services and edges represent the inclusive relations between them -- that is, a workflow is connected to a service if it consists of it. In the directed operation network, nodes are operations in services, and a directed edge represents a data link between two operations in some workflow. More details regarding the myExperiment workflow set, how networks are built and analyzed can be explored in [6].

While [6] focuses on how to build and calculate the metrics of these networks, this paper focuses on algorithms on these networks to answer Q1 and Q2. A brief summary of these algorithms is as follows. To answer Q1 we derive the frequent item sets and associations rules in the workflow-service network, and recommend relevant services in a given context (i.e., existing services in a workflow). To answer Q2 we proposed a relation-aware, cross-workflow search method to identify an operation chain which connects two services and is composed by fragments from individual workflows.

IV. Q1: PEOPLE WHO USE THESE SERVICES ALSO USE…

This section presents the ServiceMap approach to address Q1: *people who use these services also use what others?*

We adopt the well-known association rule mining method to formulate and solve this problem. Due to space limitation we only highlight the skeleton of our approach. More details regarding association rule mining can be found in [8]. In this section we used an open-source data mining tool Weka [9] to calculate frequent item sets and association rules.

Step 1: treat services as *items*.

$S=\{s_1,s_2,...,s_m\}$ is the set of services in all myExperiment workflows W (to be defined later).

Step 2: treat workflows as *transactions*.

$W=\{w_1,w_2,...,w_n\}$ is the set of workflows in myExperiment. Each workflow consists of a subset of services from S .

Step 3: calculate frequent item sets.

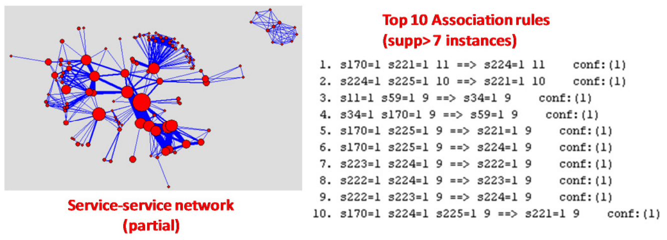
Since the number of transactions (n=347) are relatively small compared to the number of items (m=241), we do not get any large frequent set. The maximum support for any item set $X \subseteq S$ and $|X| \geq 2$ is only 5.5% (i.e., 19 items).

Step 4: calculate association rules.

This step is to find the set of all association rules, each in the form of $X \Rightarrow Y$, $X,Y \subseteq S$, $X \cap Y = \varnothing$, s.t. both $supp(X)$ and $conf(X \Rightarrow Y) \triangleq \frac{supp(X \cup Y)}{supp(X)}$ are significant enough.

The left portion of Figure 2 illustrated a portion of service-service network which is derived from the workflow-service network [6]. Nodes are services and an edge between them means that they are used together in one workflow. Node size represents how many workflows this service shows up and edge size represents how many workflows these two services show up together. It gives an intuitive view of which services are used together frequently. The right portion of Figure 2 illustrated the 10 association rules with highest confidence value and has more than 7 support workflows.

The association rules obtained can be used to suggest other relevant services usually used by peers, when a scientist has already put some services into his incomplete workflow. Due to the lack of large amount of transactions (i.e., workflows), the association rules we obtained have low support value and may not all make much sense. Despite this fact, feedback from caBIG users shows that these rules are quite informative to introduce relevant services from a large set into their experiment. The reason is that, scientific workflows are explorative and less repetitive than their business counterpart, and therefore even the association rules with low support are noteworthy. Some users told us that they are enthusiastic of the services recommended to them, in terms of generating innovative data and unexpected results.



```
 1. s170=1 s221=1 11 ==> s224=1 11    conf:(1)
 2. s224=1 s225=1 10 ==> s221=1 10    conf:(1)
 3. s11=1 s59=1 9 ==> s34=1 9    conf:(1)
 4. s34=1 s170=1 9 ==> s59=1 9    conf:(1)
 5. s170=1 s225=1 9 ==> s221=1 9    conf:(1)
 6. s170=1 s225=1 9 ==> s224=1 9    conf:(1)
 7. s223=1 s224=1 9 ==> s222=1 9    conf:(1)
 8. s222=1 s224=1 9 ==> s223=1 9    conf:(1)
 9. s222=1 s223=1 9 ==> s224=1 9    conf:(1)
10. s170=1 s224=1 s225=1 9 ==> s221=1 9    conf:(1)
```

Figure 2. Services frequently used together and the association rules

V. Q2: FIND A PATH CHAINING TWO SERVICES/OPERATIONS…

This section presents the ServiceMap approach to address Q2: *given the two or more services I want to use together, give me a path between them.*

Current scientific artifact repositories (such as BioCatalogue [1] and myExperiment) typically adopt keyword-based search. The idea is to index an entity as a vector of keywords and use the TF-IDF (term frequency-inverse document frequency) algorithm [10] to measure the weight of each keyword. Methods have recently been developed to search substructures in a tree- or graph-like structure [11] or over nested workflows [12]. These approaches suffer from two limitations. First, each result comes from a single document. For example, two workflows cannot be concatenated as a result, even by doing that you can chain two services. Second, sequential relationships cannot be established between keywords. For example, one can search for a workflow containing both services *foo* and *bar*, but cannot define the order of their appearance.

To answer Q2, we derived a directed relation between service operations. We denote this *operation network* as S' (shown in Figure 3), by examining the invocation relations among service operations. Nodes represent operations in services, and a directed edge represents a data link between two operations in an abstract workflow shown in Figure 1. S' can be seen as a directed map in which service operations are places, and workflows are routes connecting them.

Remark: S' was firstly introduced in Figure 5 of our previous network study reported in [6]. Here we illustrate a similar graph (Figure 3) for two reasons. First, as more experimental workflows are reported to myExperiment and the registered processes keep evolving, the operation network S' will keep changing. Second, we improved our workflow parsing algorithm and identified more service operations and connections that were hidden in other blocks (e.g., sub-workflow) and not previously found in [6].*A. Construction of the directed operation network*

The directed operation network S' shown in Figure 3 can be generated using the algorithm shown below. The input is the entire set of workflows stored in the myExperiment repository. S' is created incrementally: each workflow may contribute more nodes and edges into the graph.

Algorithm 1: Create operation network
Input: A repository of workflows WFR
Output: a directed operation network S'

```
1. $S' =< N, E >\leftarrow \emptyset$        //initiate the entire network
2. for each $wf \in WFR$
3.    $tmpG =< tmpN, tmpE >\leftarrow \emptyset; S \leftarrow \emptyset$
4.    find all data links in workflow $wf$
5.    for each data link $\in wf$
6.       $a \leftarrow starting\ task$;  $b \leftarrow ending\ task$
7.       if $(a/b \notin tmpN)$ then add $a/b \rightarrow tmpN$ endif
8.       add edge $(a, b) \rightarrow tmpE$
9.    end for
10.   for each $a \in tmpN$
11.      if (a is a service) then add $a \rightarrow S$ endif
12.   end for
13.   for each $a \in S$
14.      if $(a, x) \notin tmpE$ then continue endif
15.      for each $b \in S, b \neq a$
16.         len = shortest_path (a,b,tmpE)
17.         if (len≠∞) then
18.            if $(a/b \notin N)$ then
19.            then add edge $((a, b), len, wf) \rightarrow E$
20.            else update edge with $((a, b), len, wf)$
21.            endif
22.         endif
23.      end for
24.   end for
25.end for
```

Line 4 finds all data links in a workflow, each directly connecting two tasks that are building components that may or may not be external service operations. Lines 5-9 build a temporary graph, each node representing a task. Lines 10-12 identify all services invoked in the workflow.

Lines 13-24 process the temporary graph, identify the paths connecting two service operations, and add to the big service map. Line 14 checks whether a service is a starting point in any data link; otherwise, it will not be a starting point in a service path. For each possible starting service, Lines 15-16 try to find out its shortest paths to any other service. If a path exists between two services (operations), the length of the path represents the number of tasks between them. For example, if there exists a data link that directly connects two service operations, the length of the path between them is 1. If there is another task between the two service operations, the length of the path is 2. If a path is found (Step 17), Lines 18-20 add the path together with its length into the big service map.

Note that S' records multiple paths between a pair of operation nodes, each representing an operation chain (may involving multiple intermediate, local, non-service

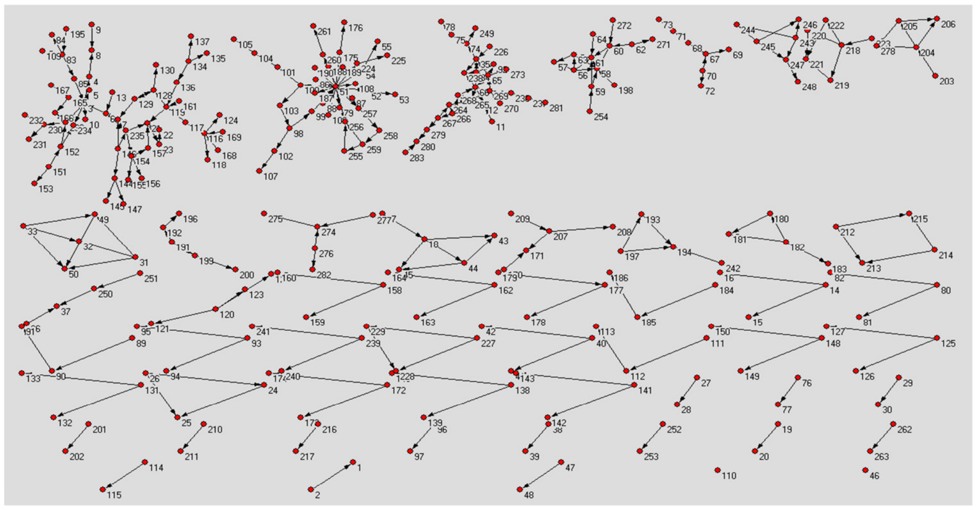


Figure 3. S' as a directed network: service operations are nodes, and a directed edge represents a data link between two operations in an abstract workflow.

steps between them) within one specific workflow. The rationale is that, a domain scientist may select a specific path because the corresponding workflow is created by her collaborator.

In general, any shortest path algorithm can be applied in Line 15 to find paths between a pair of service operations, such as the well-known Dijkstra algorithm. Since our experiment exposed the sparseness feature (will discuss in Section 5), we applied an enhanced version of the Dijkstra algorithm for a higher performance, by storing the graph in the form of adjacency lists and using a Fibonacci heap as a priority queue.

B. Motivation for relation-aware and cross-workflow search

We propose to use network S' and explore a relation-aware and cross-workflow search technique. Before diving into details, let's take an example to explain how S' supports such a search. A typical question in bioinformatics is that, "Given a DNA sequence, can I first find similar ones from *WU-BLAST* and then compare them with those from *ClustalW2*?" (*WU-BLAST* and *ClustalW2* are two popular sequence alignment services.) To answer this question, we can leverage relation-aware search to find candidate workflows that start from some operation in *WU-BLAST* and end with some operation in *ClustalW2*. The question can be thus rewritten into the following query:

search workflow where WU-BLAST → ClustalW2

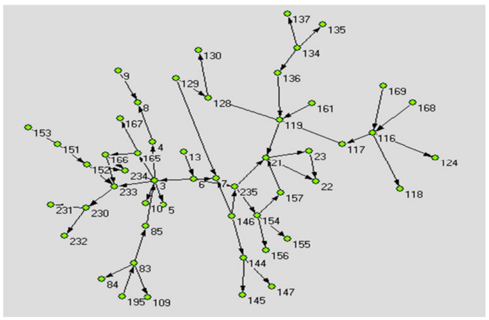


Figure 4. Obtain a service chain between two operations

It is quite likely that, in the repository there is not a single workflow which contains an operation chain starting from *WU-BLAST* and ends at *ClustalW2*. In this case the result should be a new workflow concatenating snippets from several existing ones. We use the largest cluster in S' to demonstrate this idea. This cluster, which is in the top-left of Figure 3, is shown in Figure 4 with the name of each node (i.e., operation) in it. Operations 116, 117, 168 and 169 belong to service *WU-BLAST*; 128 and 130 belong to service *ClustalW2*. Two paths <169,116,117,119,128,130> and <168,116,117,119,128, 130> are two candidate paths satisfying the search criterion. Based on scientist-side context (e.g., with the DNA sequence at hand), the first candidate workflow will be suggested (the data format matches operation 169's input. The resulting routine actually is a concatenation of three snippets each of which from a myExperiment workflow.

116	WSWUBlast.wsdl#runWUBlast
117	WSWUBlast.wsdl#getIds
119	WSDbfetch.wsdl#fetchBatch
128	WSClustalW2.wsdl#runClustalW2
130	WSClustalW2.wsdl#checkStatus
168	WSWUBlast.wsdl#getPrograms
169	WSWUBlast.wsdl#getDatabases

This example demonstrates that S' allows us to discover global relations spread in multiple workflows and originally not easy to identify. Our experience working with caBIG community shows that this feature is quite useful for scientists to explore best practices from multiple colleagues and combine their experiment snippets into a more comprehensive one.

C. Method for relation-aware and cross-workflow search

This sub-section describes the method for relation-aware and cross-workflow search. For simplicity Q2 is formulated as follows: given two operations o_i, o_j, a set of workflows W and the derived operation network S', how to find a path in S' that connects o_i, o_j and meets a certain criterion, e.g., this path should cross least number of workflows.

The across-least-workflow criterion is reasonable because each time when two snippets from two workflows are to be concatenated, additional tuning such as data transformation and security enforcement are needed. Therefore a path which crosses less number of workflows is more desirable. Again if we make this analogy that operations are stops in a public transportation system and a workflow is a bus/subway route connecting multiple stops, we prefer a path between two stops which crosses less number of routes, i.e., with less transfer overhead.

$$R_i = A_i^* = A_i + A_i^2 + ... + A_i^{m_i - 1}$$

$$R^{k+1} = R^k \bullet R$$

$$T = [t_{ij}]_{m \times m} : t_{ij} \triangleq \{\min k \in [1, m] \text{s.t.} r_{ij}^k > 0\}$$

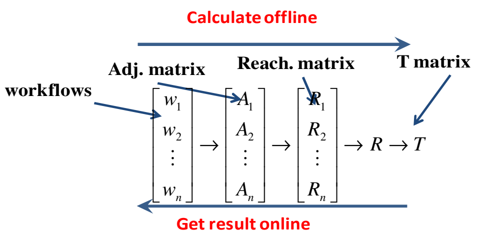


Figure 5. Obtain least transfer route between two operations

Figure 5 is a summary of the approach proposed in this section. Three matrixes, i.e., adjacent (A), reachability (R) and transfer (T) matrixes are calculated offline and sequentially (their definition are given later). When finding a path between two operations, from matrix T we know how many transfers is needed; by referring back from matrix T to R and A we obtain these paths.

Step 1: calculate adjacent matrix

$W = \{W_1, W_2, ..., W_n\}$ is the set of workflows we extracted from myExperiment.

$W_i = (O_i, E_i)$ in which O_i is the set of (service) operations in W_i and E_i is the set of edges between these operations.

$A_i : O_i \times O_i \rightarrow \{0,1\}$ is the *adjacent matrix* of W_i. That is, given $o_{ik}, o_{ij} \in O_i$, element $[k, j]$ of A_i is defined as:

$$A_{ikj} = \begin{cases} 1 \text{ if } o_{ik} \text{ has a link to } o_{ij} \text{ in } W_i \\ 0 \text{ otherwise} \end{cases}$$

A_i can be directly obtained from W_i. Based on adjacent matrix we can derive the reachability matrix of each workflow.

Step 2: calculate reachability matrix

$R_i : O_i \times O_i \rightarrow \{0,1\}$ is the *reachability matrix* of W_i.

Given $o_{ik}, o_{ij} \in O_i$, element $[k, j]$ of R_i is defined as:

$$R_{ikj} \triangleq R_i(o_{ik}, o_{ij}) \triangleq \begin{cases} 1 \text{ if } o_{ik} \text{ can reach } o_{ij} \text{ in } W_i \\ 0 \text{ otherwise} \end{cases}.$$

$$\forall o \in O_i, R_i(o, o) \triangleq 0$$

R_i cannot be directly obtained from W_i, instead it can be calculated from matrix A_i^*:

$$A_i^* \triangleq A_i + A_i^2 + .. + A_i^{m_i - 1} \ (m_i \text{ is the cardinality of } A_i)$$

$$R_{ikj} = \begin{cases} 1 \text{ if } A_{ikj}^* > 0 \\ 0 \text{ otherwise} \end{cases}$$

R_i represents the reachability relations between any two operations in W_i, and by aggregating them we can examine the reachability relations among operations in a global, cross-workflow perspective, i.e., in S'.

Given the workflow set $W = \{W_1, W_2, ..., W_n\}$ and the operation set $O = \{o_1, o_2, ..., o_m\}$, we now combine $\{R_1, R_2, ..., R_n\}$ into an aggregated relation $R = [r_{ij}]_{m \times m}$:

$r_{ij} \triangleq \sum_k \{R_{kij} = 1\}$, i.e., r_{ij} equals the number of workflows in which o_i can reach o_j, and $r_{ii} \triangleq 0$.

We then calculate the n^{th} power of R:

$$R^n = \left[r_{ij}^n\right]_{m \times m} \text{ and } r_{ij}^n \triangleq \sum_{k=1}^{n} r_{ik}^{n-1} r_{kj}$$

Therefore, r_{ij}^n equals the number of times o_i can reach o_j by traversing n workflows, i.e., transferring $n-1$ times. Now we know for each given operation pair, the existence of a directed chain which is across a certain amount of workflows and connects these two operations.

Step 3: calculate transfer matrix

At the beginning of this section we emphasize that we want a service chain *across least workflows*, like passengers want an itinerary across least routes (i.e., with least transfers). In order to achieve that goal we need to introduce another transfer matrix T.

Given m operations and n workflows:

$T = [t_{ij}]_{m \times m} : t_{ij} \triangleq \{\min k \in [1, n] \text{s.t. } r_{ij}^k > 0\}$

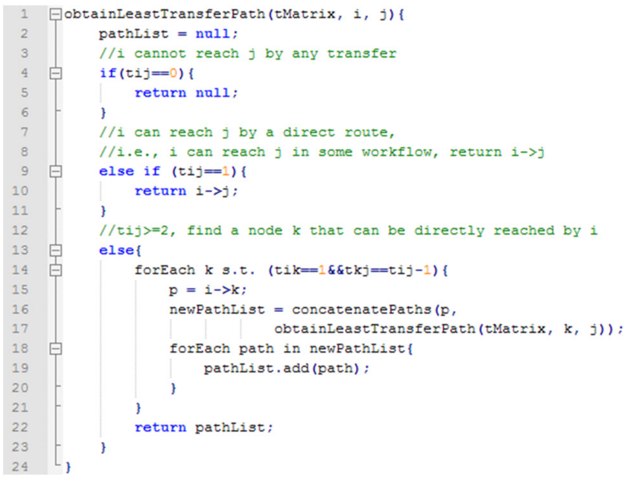
T reveals the least-transfer distances between two operations. By definition, $t_{ij} - 1$ is the number of least transfers through which o_i can reach o_j. If $t_{ij} = 0$ then there is not a path between them across the available workflows.

Step 4: calculate transfer paths

For $\forall t_{ij} = k \geq 2$, at least one transfer is needed for any path from o_i to o_j, i.e., we can find a least-transfer path $o_i \rightarrow o_{q_1} \rightarrow \cdots \rightarrow o_{q_{k-1}} \rightarrow o_j$, such that:

$$t_{iq_1} = t_{q_1 q_2} \cdots = t_{q_{k-2} q_{k-1}} = t_{q_{k-1} j} = 1$$

Such a path (or paths) can be found using the recursive algorithm shown in Figure 6.



```
1  obtainLeastTransferPath(tMatrix, i, j){
2       pathList = null;
3       //i cannot reach j by any transfer
4       if(tij==0){
5           return null;
6       }
7       //i can reach j by a direct route,
8       //i.e., i can reach j in some workflow, return i->j
9       else if (tij==1){
10          return i->j;
11      }
12      //tij>=2, find a node k that can be directly reached by i
13      else{
14          forEach k s.t. (tik==1&&tkj==tij-1){
15              p = i->k;
16              newPathList = concatenatePaths(p,
17                              obtainLeastTransferPath(tMatrix, k, j));
18              forEach path in newPathList{
19                  pathList.add(path);
20              }
21          }
22          return pathList;
23      }
24 }
```

Figure 6. Obtain least transfer paths between two operations

Line 1: given T and $< o_i, o_j >$, find a service chain between o_i and o_j with least transfer.

Line 2: initiate the result path set to null.

Line 4-6: $t_{ij} = 0$ means *i* cannot reach *j* by any transfer.

Line 9-11: $t_{ij} = 1$ means *i* can reach *j* without transfer; find the workflow in which *i* can reach *j* and get the service chain in it.

Line 13-21: $t_{ij} \geq 2$ means i can reach j with at least one transfer; find an immediate node k that i can reach and connect it with a least-transfer path from k to j.

Line 24: return the result path list.

VI. AN EMPIRICAL STUDY

Given the matrix-based approach described in 5.3, we calculated the least-transfer distances and transfer paths between any pair of service operations invoked in the workflow set, which are registered in the myExperiment repository. The dataset was taken from myExperiment on Aug-23-2010 and it contains 347 workflows, 241 services and 283 operations. Figure 7 is the histogram showing the least-transfer distances between any pair of nodes representing the 283 operations in S' (i.e., 283*283 = 80,089 node pairs). 375 pairs of operations can reach each other without any transfer, i.e., they are connected within a single workflow. 147 operation pairs are reachable via 1 transfer; and 61 and 14 pairs are reachable via 2 and 3 transfers, respectively. Only 2 pairs of operations are reachable via 4 transfers; and there is no path with more than 4 transfers between any operations.

In contrast to a public transportation system, the reachability among the operations is obviously sparse: only 375+147+61+14+2=599 out of 80,089 operation pairs are reachable (about 0.75%); if two nodes are not reachable within 4 transfers, they are not reachable at all. The sparseness is due to two major reasons. First, bioinformatics services/operations cannot be arbitrarily connected, because they have different usage scenarios by nature. Second, the services in the myExperiment workflows largely function individually rather than collaboratively [6]. Although the myExperiment workflow set only illustrates a subset of the usage of the bioinformatics services, our experiment does reveal the necessity of increasing the reusability and collaboration among bioinformatics services.

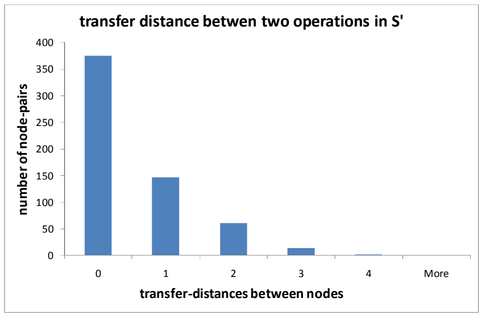


Figure 7. Least-transfer-distance histogram of myExperiment workflows

VII. RELATED WORK

Our presented work is closely related to three categories of research: automatic service composition, artifact recommendation in software engineering and scientific workflow, and network analysis in software engineering and workflow.

A. Automatic service composition

Various automatic service composition techniques have been developed to discover relevant services and compose them in a proper sequence, as discussed in the survey [13]. These methods can yield good results when candidate services provide comprehensive metadata (such as input/output, pre/post conditions, and QoS) and/or the composition problem can be translated into a good formalism, like optimization, AI-based planning, and process algebra. In bioinformatics, however, many services are widely used without metadata and formal models. Meanwhile, online workflow repositories (such as myExperiment) allow scientists to share successful experimental routines that contain best practices to compose services. Based on this observation, we have adopted a network-based approach to analyze the usage patterns of the biomedical services and provide recommendations based on empirical workflows.

B. Artifact Recommendation

Artifact discovery and reuse has been investigated in software engineering [14, 15], with an emphasis on mining patterns from software repositories and recommending software artifacts. In the area of scientific workflow, there are already many systems including Kepler [16], Taverna [4], Pegasus [17], and Swift [18]. The popular business workflow language BPEL has also been adopted [19]. More recently, patterns from past usage data are summarized to predict the most likely next-step in building visualization pipelines [20], and case base reasoning is also used to find a similar workflow and use it to suggest the next component to use [21]. Our ServiceMap differentiates from those approaches by providing suggestions which are relation-aware and across multiple workflows.

C. Network analysis

Researchers have recently started to build social networks to support scientific application sharing. myExperiment and BioCatalogue are two examples maintaining social networks for life science workflows and services, respectively. In software engineering, Codebook [22] builds a directed graph connecting programmers to reusable work artifacts from software repositories. Within the directed graph, a regular language reachability algorithm is used to discover any transitive connections.

In ServiceMap, we have adopted a matrix-based approach similar to [23] that calculates transfer routes in a public transportation system. ServiceMap is inspired by the analogy between service-based workflows and transportation network, but our approach addresses many challenges unique to service composition and validated by empirical data from myExperiment.

In business process management (BPM) field, a framework named TomTom4BPM [24, 25] adopts process mining technique for various purposes, such as comparing the actual process execution with pre-modeled ones and dynamically navigating during process exceptions. In contrast, our work focuses more on suggesting new service/workflows based on existing ones during *construction time*, while TomTom4BPM concentrates on mining process logs to better navigate execution during *run-time*.

VIII. CONCLUSIONS

As a continuous effort of our network analysis on the myExperiment [6], we established a *ServiceMap* framework. Its association rule mining and matrix-based searching algorithms aim to provide a GPS-like support serving two types of scientific service searching queries: one is to find services frequently used together within a given service set; the other one is to identify operation chains connecting services from their past usages.

This work is an integral part of a larger framework **CASE** (information **C**ollection, **A**nnotation, **S**earch, and r**E**commendation), which is under development aiming to support scientific artifact reuse [6]. In the future, we plan to integrate the matrix-based approach with other path planning methods, such as AI-based ones. Meanwhile, we currently focus on recommending services and service chains by looking at the task-flow in workflows. We plan to explore a complementary *data-driven* approach, studying data flows and their connectivity across multiple workflows using a Petri-net based decomposition approach developed earlier in [26].

ACKNOWLEDGEMENTS

We thank the caBIG community for the motivating scenarios and feedbacks to our work. The work is partially supported by the National Cancer Institute, the National Institutes of Health under contract N01-CO-12400, and the National Science Foundation under grant IIS-0959215.

REFERENCES

[1] J. Bhagat, F. Tanoh, E. Nzuobontane, T. Laurent, J. Orlowski, M. Roos, K. Wolstencroft, S. Aleksejevs, R. Stevens, S. Pettifer, R. Lopez, and C.A. Goble, "BioCatalogue: a universal catalogue of web services for the life sciences", *Nucl. Acids Res.*, May 19, 2010, 2010: pp. gkq394.

[2] I.J. Taylor, E. Deelman, D.B. Gannon, and M. Shields, *Workflows for e-Science: Scientific Workflows for Grids*, 2007: Springer-Verlag.

[3] A.C. Von Eschenbach and K. Buetow, "Cancer Informatics Vision: caBIG™", *Cancer Informatics*, 2006, 2: pp. 22.

[4] T. Oinn, M. Greenwood, M. Addis, M.N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M.R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, "Taverna: lessons in creating a workflow environment for the life sciences", *Concurrency and Computation: Practice & Experience*, 2006, 18(10): pp. 1067-1100.

[5] W. Tan, R. Madduri, A. Nenadic, S. Soiland-Reyes, D. Sulakhe, I. Foster, and C. Goble, "CaGrid Workflow Toolkit: A taverna based workflow tool for cancer grid", *BMC Bioinformatics*, 2010, 11(1): pp. 542.

[6] W. Tan, J. Zhang, and I. Foster, "Network Analysis of Scientific Workflows: A Gateway to Reuse", *IEEE Computer*, 2010, 43(9): pp. 54-61.

[7] C.A. Goble, J. Bhagat, S. Aleksejevs, D. Cruickshank, D. Michaelides, D. Newman, M. Borkum, S. Bechhofer, M. Roos, P. Li, and D.D. Roure, "myExperiment: a repository and social network for the sharing of bioinformatics workflows", *Nucleic Acids Research*, 2010(38(Web Server issue)): pp. W677-W682.

[8] P.N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, 2006: Pearson Addison Wesley Boston.

[9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, "The WEKA data mining software: An update", *ACM SIGKDD Explorations Newsletter*, 2009, 11(1): pp. 10-18.

[10] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval", *Information processing & management*, 1988, 24(5): pp. 513-523.

[11] H. He, H. Wang, J. Yang, and P. Yu, "BLINKS: ranked keyword searches on graphs", in *ACM SIGMOD international conference on Management of data*. 2007, ACM. pp. 305-316.

[12] Z. Liu, Q. Shao, and Y. Chen, "WISE: Searching Workflow Hierarchies", in *36th International Conference on Very Large Data Bases*. 2010: Singapore

[13] S. Dustdar and W. Schreiner, "A survey on web services composition", *International Journal of Web and Grid Services*, 2005, 1(1): pp. 1-30.

[14] T. Xie, S. Thummalapenta, D. Lo, and C. Liu, "Data Mining for Software Engineering", *IEEE Computer*, 2009, 42(8): pp. 55-62.

[15] R. Martin, W. Robert, and Z. Thomas, "Recommendation Systems for Software Engineering", *IEEE Software*, 2010, 27: pp. 80-86.

[16] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, and Y. Zhao, "Scientific Workflow Management and the Kepler System", *Concurrency and Computation: Practice & Experience*, 2005.

[17] E. Deelman, "Pegasus: A framework for mapping complex scientific workflows onto distributed systems", *Scientific Programming*, 2005, 13(3): pp. 219-237.

[18] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, I. Raicu, T. Stef-Praun, and M. Wilde, "Swift: Fast, Reliable, Loosely Coupled Parallel Computation", in Proceedings of *2007 IEEE Congress on Services*, 2007, pp. 199-206.

[19] B. Wassermann, W. Emmerich, B. Butchart, N. Cameron, L. Chen, and J. Patel, "Sedna: A BPEL-Based Environment for Visual Scientific Workflow Modeling", in *Workflows for E-science: Scientific Workflows for Grids*, I.J. Taylor, E. Deelman, D.B. Gannon, and M. Shields, Editors, 2007, Springer-Verlag, pp. 428-449.

[20] D. Koop, C.E. Scheidegger, S.P. Callahan, J. Freire, and C.T. Silva, "VisComplete: Automating Suggestions for Visualization Pipelines", *IEEE Transactions on Visualization and Computer Graphics*, 2008, 14: pp. 1691-1698.

[21] E. Chinthaka, J. Ekanayake, D. Leake, and B. Plale, "CBR Based Workflow Composition Assistant", in *2009 Congress on Services -- I*. 2009, IEEE Computer Society. pp. 352-355.

[22] A. Begel, K.Y. Phang, and T. Zimmermann, "Codebook: discovering and exploiting relationships in software repositories", in *32nd ACM/IEEE International Conference on Software Engineering*. 2010, ACM: Cape Town, South Africa. pp. 125-134.

[23] C.L. Liu, T.W. Pai, C.T. Chang, and C.M. Hsieh, "Path-planning algorithms for public transportation systems", in Proceedings of *14th IEEE Intelligent Transportation Systems*, 2001, IEEE, pp. 1061-1066.

[24] W. van der Aalst, "TomTom for Business Process Management (TomTom4BPM)", in Proceedings of *21st International Conference on Advanced Information Systems Engineering (CAiSE)*, 2009, Springer, pp. 2-5.

[25] W.M.P. Aalst, "Using process mining to generate accurate and interactive business process maps", in Proceedings of *International Business Information Systems Workshops* 2009, Springer, pp. 1-14.

[26] W. Tan, Y. Fan, M. Zhou, and Z. Tian, "Data-Driven Service Composition in Enterprise SOA Solutions: A Petri Net Approach", *IEEE Transactions on Automation Science and Engineering*, 2010, 7(3): pp. 686-694.