

Computational Analysis of Clinical Practice Guidelines

**Development of a software suite and document standard for storage
and analysis of care maps**

by

Varun Ramraj

B.Sc., University of British Columbia, 2008

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Bioinformatics Graduate Program)

The University Of British Columbia
(Vancouver)

September 2010

© Varun Ramraj, 2010

Abstract

Clinical Practice Guidelines (CPGs) guide optimal utilization of clinical delivery of health care through evidence-based medicine, where care procedures are rigorously evaluated and improved through the examination of evidence. Care mapping is the technique of using flowcharts to graphically capture CPGs as discrete, actionable steps. Health professionals can create and use care maps to expedite and ensure excellence in optimal process workflow in patient care. Analysis of care maps would provide insight into similarities and differences in care procedures. However, quantitative analysis of care maps is difficult to perform manually, and becomes impossible as the set of care maps for comparison increases. Computational methods could be employed to obtain the required quantitative data, but current document standards for developing, sharing and visualizing care maps are not rigorous enough for computational analysis to take place. By using Bioinformatics approaches, we can solve these problems. Firstly, we can develop a standard care map file format for electronic storage. Systems Biology Markup Language (SBML), a document format used to describe biological pathways, can be used to develop the required file format. This method works because care maps are notionally very similar to biological pathways. It allows use of multiple alignment algorithms (traditionally used to align and cluster biological pathways) with these transformed care maps in order to derive quantitative data. This project involved the development of a software suite that is able to generate care maps in the SBML format and align them using an existing global multiple pathway alignment algorithm. It is part of a larger project that examines efficacy of CPGs. This would allow for two important studies to be conducted: a breadth study across multiple

Emergency Departments (EDs) and a longitudinal study over time within a single ED to see how it has been able to implement and adapt to the CPGs. By utilizing Bioinformatics approaches in care mapping, two important objectives were realized: the creation of a document standard for care maps, and computational comparison and contrast of CPGs. This opens up the exciting new field of Translational Informatics, which applies existing Bioinformatics concepts to e-Health, e-Medicine and Health Informatics.

Contents

Abstract	ii
Contents	iv
List of Tables	vii
List of Figures	viii
List of Abbreviations	x
Acknowledgments	xiv
1 Introduction	1
1.1 Structured Care Approaches in Health Care - A Primer	1
1.1.0.1 Protocols	2
1.1.0.2 Care Maps	3
1.1.0.3 Algorithms	4
1.1.0.4 Decision Analysis	7
1.1.1 Clinical Practice Guidelines - An Evidence Based Structured Care Approach (SCA)	8
1.1.2 CPGs as an Aid to Electronic Decision Support	12
1.1.3 The Importance of Quantitative Information	14
1.2 Making CPGs Viable for Computational Analysis - Project Objective and Approach	15
1.2.1 Types of Analyses	15
1.2.2 Considerations in Building the CPG Document Standard	16

1.2.2.1	Institution-specific Customization of Care Maps	19
1.2.2.2	Scope of a Care Map	19
1.2.2.3	Care Map Sharing and Delivery Methods	20
1.3	Considerations for Computational CPG Analysis	20
1.4	Adaptation of Biological Pathway Alignment Concepts	21
1.5	Context of this Project	22
1.6	Project Goals and Anticipated Outcome	24
2	Methods	25
2.1	Translation - Choosing SBML to Design the Care Map Standard .	25
2.2	Analysis - Pathway Alignment Approach	27
2.2.1	Notes on the IsoRank Pathway Alignment Algorithm . . .	27
2.2.1.1	Interaction Data	28
2.2.1.2	Similarity Data and Vocabulary Control	28
2.3	Implementation - Software Design and Development	31
2.3.1	Cross-Platform Compatibility	32
2.3.2	Modularity	32
2.3.3	The <code>carmat</code> Module and Manual Care Map Curation . .	34
2.3.3.1	<code>carmat</code> Module Design	34
2.3.3.2	The Vocabulary Map	35
2.3.3.3	Manual Care Map Vocabulary Curation	38
2.3.4	The <code>aligner</code> module - A Cross-Platform IsoRank Imple- mentation	38
2.3.5	The Chequers Graphical User Interface (GUI)	40
2.3.6	License	41
3	Results, Discussion and Future Directions	42
3.1	Results	42
3.1.1	Viability of a Curated SBML-based Care Map Document Standard	42
3.1.2	Alignment Results	43
3.2	Discussion and Implications	46

3.2.1	Notes on Extensible Markup Language (XML) and Sharability	47
3.2.1.1	The SBML Layout and Render Extensions	48
3.2.1.2	Real-Time Collaboration and Revision Control	49
3.2.2	IsoRank’s Viability in this Use Case	50
3.2.2.1	Big Hammers for Small Nails	52
3.2.3	Musings on Granularity	52
4	Future Directions	54
4.1	Comparison of Multiple Care Maps	54
4.2	Scoring Matrices	54
4.3	Improving the Document Standard’s Sharability	55
4.3.1	Thoughts on Mobile Chequers	57
4.4	Vocabulary Control and General Aesthetics	58
4.5	Future Clinical Trials with Chequers	59
5	Conclusion	61
	Bibliography	63
A	A More Formal Definition of Care Maps	69
A.1	A Basic Care Map	69
A.1.1	Incorporation of an Algorithm	70

List of Tables

Table 1.1	The types of data obtainable from computational analysis of CPGs for a given procedure or intervention. Each analysis method provides specific types of data based on the number of care facilities considered in the analysis.	16
-----------	---	----

List of Figures

Figure 1.1	The SCA hierarchy. The care map is a graphical structure. It is composed of protocols at its core, and can be enhanced by algorithms or decision analyses.	2
Figure 1.2	A generic sepsis care map which incorporates two algorithmic decision steps or branch points (identified by the letter “A”). The care map outlines tasks to be conducted by different care givers, thus, it amalgamates different care protocols.	6
Figure 1.3	The positive feedback cycle generated by the use of evidence-based medicine. 1) The CPG defined in Figure 1.1 . 2) Every level above the protocol is 3) analyzed for potential improvements which are then 4) inserted into the protocols. 5) The modified protocols are re-implemented into the CPG.	11
Figure 1.4	The basic components of a care map showing branch points and the definitions of “upstream” and “downstream.”	18
Figure 2.1	Some of our controlled vocabulary.	30
Figure 2.2	A class diagram showing Chequers’ three main modules and classes that make up each module. The <code>chequersGUI</code> class integrates functionality from <code>aligner</code> and <code>carmat</code> and presents it to the user in graphical format.	33

Figure 2.3	The use of a hash map as an intermediary between the vocabulary and the SBML document satisfies SBML’s stringent requirements for formatting of the data in the document. Since the hash value is generated from the vocabulary, applying the hash function in reverse on the hash number will return the vocabulary sentence.	37
Figure 3.1	A portion of the same care map, curated with controlled vocabulary and rendered with the SBML layout tool directly into Scalable Vector Graphics (SVG) format.	43
Figure 3.2	Results of aligning Hospital 1’s and 2’s care maps as produced by <code>aligner</code> . In total, using the input constants that we did ($\alpha = 0.6, \beta_2 = 0.8$), five clusters were generated.	45

List of Abbreviations

AI artificial intelligence.....	12
API Application Programming Interface, a set of software libraries provided for integration into new software.....	26
Atom Atom Syndication Format, an alternative to Really Simple Syndication (RSS) for web feed content.....	57
BN Bedside Nurse	69
BIDMC Beth Israel Deaconess Medical Centre, located in Boston, Massachusetts 10	
BLAST Basic Local Alignment Search Tool.....	28
CPG Clinical Practice Guideline, an evidence-based medical guideline for patient management and process workflow that depends on the patient’s condition and symptoms. Described in narrative form. A care map is a graphical representation of this narrative.....	8

ED Emergency Department	3
EHR Electronic Health Record	46
EMR Electronic Medical Record	12
EBCA Evidence-based Clinical Algorithm	8
FOSS Free and Open Source Software	31
FTP File Transfer Protocol	47
gcc GNU Compiler Collection	31
GO Gene Ontology	29
GPL GNU General Public License	41
GUI Graphical User Interface	21
HTTP Hypertext Transfer Protocol	47
JPEG Joint Photographic Experts' Group	13
LAN Local Area Network	49

MUST Multiple Urgent Sepsis Therapies, a protocol developed at Beth Israel Deaconess Medical Centre in Boston, Massachusetts.	10
NVA non-value-added	7
OSS Open Source Software	31
PDF Portable Document Format	13
PS Postscript document	35
RSS Really Simple Syndication, a method of delivering frequently updated web content to browsers or mobile devices.	57
SBML Systems Biology Markup Language, an XML-based markup language for describing biological pathways.	22
SCA Structured Care Approach, an approach towards care that could be a protocol, clinical pathway, decision analysis or algorithm.	1
SCM Software Configuration Management, also known as Source Code Management, a method of maintaining software source code.	49
SVG Scalable Vector Graphics	ix
TAR Tape Archive	35

TN Triage Nurse	69
URL Uniform Resource Locator, used to describe a means for obtaining some resources on the World Wide Web.	57
VA value-added	7
OWL Web Ontology Language, a knowledge representation language for creating and using ontologies.	29
XML Extensible Markup Language, a flexible standard for creating custom structured document formats.	25

Acknowledgments

I wish to thank my supervisor, Dr. Kendall Ho, and my supervisory committee members, Dr. Charles Krasic and Dr. Arvind Gupta for their support and enthusiasm in a project that tries hard to push the boundaries of classical Bioinformatics. I would also like to thank Frank Bergmann at the University of Washington for his help with the Systems Biology Markup Language, the `libsbml` C++ library, and the Systems Biology Workbench software. Lastly, a big thanks to Jan Manuch of MITACS at UBC for his clear explanation and help with my implementation of the IsoRank algorithm.

Chapter 1

Introduction

1.1 Structured Care Approaches in Health Care - A Primer

An important organizational concept of health care management is the mapping of patient management workflow known as Structured Care Approach (SCA) (Cole et al., 1996; Advani et al., 1998; O'Neill, 2000; Boxwala et al., 2001; Picard et al., 2006). The goal of an SCA is to document and implement standardized care guidelines in order to provide the best care in as efficient a manner as possible. As its name implies, SCA characterizes clinical care provision using a defined structure, articulating a series of steps taken by care givers when treating patients. SCAs can be used to guide clinicians and nurses in making complex decisions tailored to individual patient care while still maintaining the standards set out by their respective governing bodies and health care institutions. They also can promote transparency in the critique of health care facilities in their different steps of care provision through outcome measurement, outcome management and measurement of variance in care compared to other facilities.

SCAs comprise several conceptually linked approaches including protocols, care maps, algorithms and decision analyses, summarized in **Figure 1.1**.

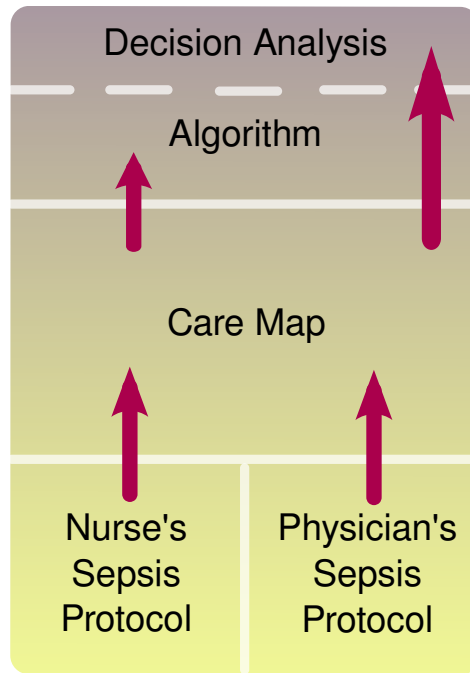


Figure 1.1: The SCA hierarchy. The care map is a graphical structure. It is composed of protocols at its core, and can be enhanced by algorithms or decision analyses.

1.1.0.1 Protocols

Protocols form the basic building blocks of an SCA (**Figure 1.1**); they are comprehensive instructions or checklists used as references by caregivers to conduct a procedure or treat a specific malady. They contain detailed information and suggestions about the steps that should be carried out by a particular care giver. For example, nurses at the emergency department might refer to a protocol to decide how sick a patient is and how soon he should be seen by a doctor - a process known as **triage**. The protocol is described in great detail in order to provide the nurses with as much information as possible to perform their duties up to the standards set out by the protocol to correctly direct the patient to be appropriately treated in a timely manner based on severity of his illnesses. Similarly, a physician can

be effectively guided by protocols to ensure proper care based on best practices (O'Neill, 2000).

1.1.0.2 Care Maps

Care facilities employ multiple types of caregiver. In Emergency Departments (EDs) for example, physicians, triage nurses, bedside nurses and other hospital staff work together in order to provide care to the patients. While the different staff members apply their own professional expertise and approaches to caring for the patients, their individual actions should cohesively work towards the overall care of the patients. A clinical **care map** is an attempt to capture and summarize all the different staff members' treatment procedures into a united pathway to characterize the multidimensional care that the patients receive, either for a particular illness or in a particular clinical context such as the overall patient flow in an ED, or both simultaneously (Cole et al., 1996; Advani et al., 1998; O'Neill, 2000; Boxwala et al., 2001; Picard et al., 2006). For example, an "Emergency Department Sepsis Care Map" describes how a patient suffering from infection would be treated by the physicians, nurses and other allied health professionals during the course of said patient's stay in the ED. Thus, a care map is an amalgamation of multiple protocols that describe the same ailment care procedure (Cole et al., 1996; O'Neill, 2000).

Care maps are often developed for cases when care facilities experience a large volume of patients expressing the same high-risk ailment and where a team of care givers from many disciplines must be assembled to handle the situation (Lovejoy et al., 1997; Golden and Ratliff, 1997). For example, sepsis patients are often admitted to EDs in high volumes and sepsis requires many types of care givers (physicians, triage nurses, bedside nurses) to work together to provide optimum patient care. In such cases, proper care could become challenging if each care giver were following a discrete protocol that did not harmonize with each other's protocols, thereby failing to work as a cohesive team. A care map that integrates multiple care givers' protocols however, would allow nurses and physicians to prioritize their tasks, perform them systematically and work synergistically in a manner conducive to optimal patient outcome.

Figure 1.2 shows a generic care map that contains some of the fundamental steps required in sepsis treatment at an ED. It is apparent that this care map outlines sepsis care tasks to be conducted by different types of personnel. It is an implementation of a standardized approach, whereby, if a patient were to walk into this ED with sepsis symptoms, this care map could be followed to administer the proper care. However, each patient is unique and has his or her own individual needs. The challenge in care mapping is to develop a roadmap of care that is broad enough to encompass all anticipated types of patients, yet flexible and targeted enough to accommodate an individual set of patient symptoms and needs. In order to encompass care pathways for many types of patients, care maps need to be enhanced beyond their rudimentary form using **algorithms** (O’Neill, 2000; Lovejoy et al., 1997; Golden and Ratliff, 1997).

Section A in the **Appendix** uses mathematical notation to describe a basic care map.

1.1.0.3 Algorithms

As stated above, all protocols and care maps, as currently defined, have a drawback. They do not make any assumptions about individual patients; they are specific about the procedures and roles of the care givers, but are patient-agnostic. Thus, from the vantage point of rigid care maps, every patient that walked into an ED would be treated in exactly the same way regardless of individual parameters or contexts (O’Neill, 2000; Lovejoy et al., 1997; Golden and Ratliff, 1997). Therefore, a key struggle in creating SCAs rests in building care maps generic enough to account for every permutation of patient symptoms for a given malady, yet be encompassing enough to guide the care givers through proper procedural steps and systematic structures.

Algorithms address this challenge by providing a way to amalgamate portions of care maps together through the use of decision points or branch points. They resemble computational algorithms through the use of `if-then-else` statements or Boolean (true-false) logic. This kind of branching logic allows care givers to

use algorithms to tailor care methods based on patient parameters such as symptoms or past medical conditions (O’Neill, 2000). It is important to note that, rather than being graphically distinct from care maps, algorithms are enhancements to the care maps themselves. For instance, **Figure 1.2** is a care map that contains two decision-making steps (identified by the letter “A”), demonstrating the use of algorithms to enhance but still be an integral element of it.

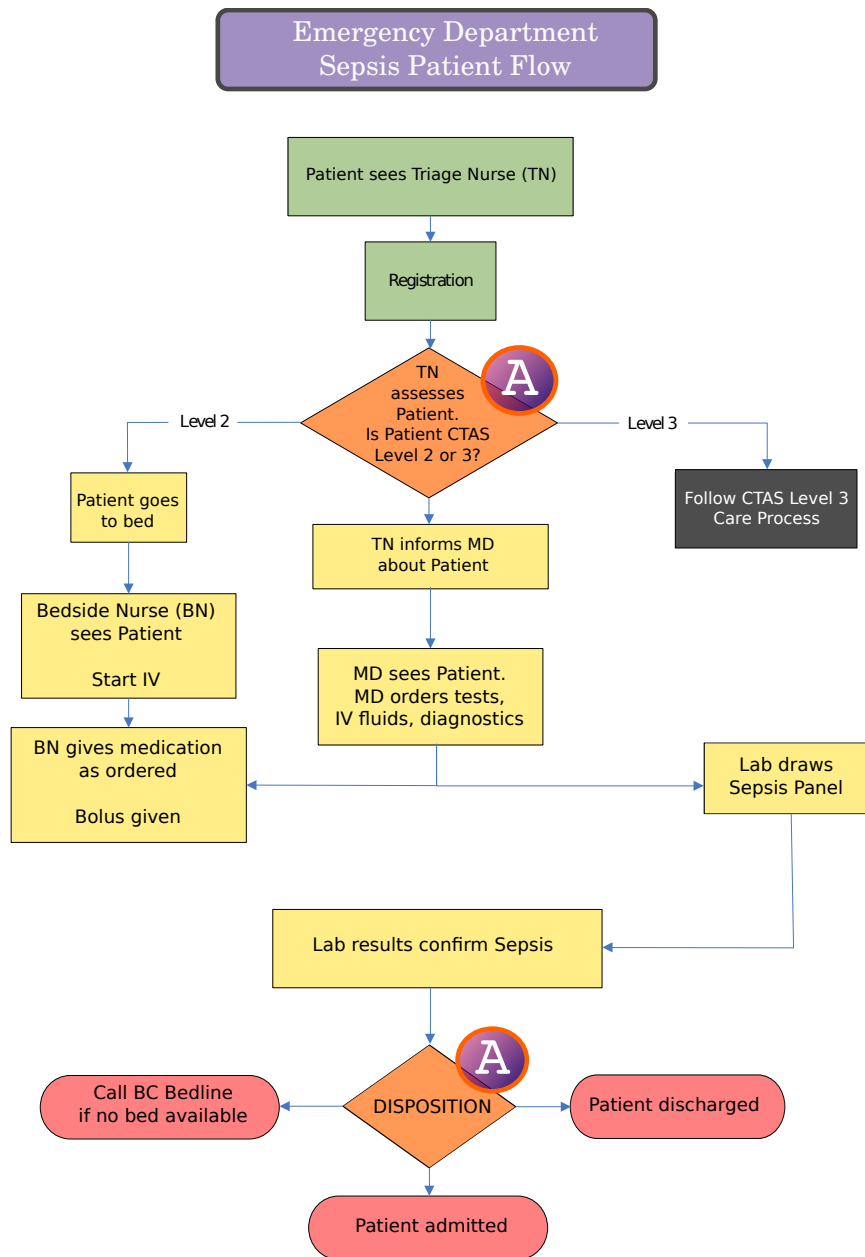


Figure 1.2: A generic sepsis care map which incorporates two algorithmic decision steps or branch points (identified by the letter “A”). The care map outlines tasks to be conducted by different care givers, thus, it amalgamates different care protocols.

Section A in the **Appendix** is a more detailed mathematical model of a care map with the incorporation of a simple algorithm in order to compare and contrast between care maps alone and care maps with algorithmic components.

1.1.0.4 Decision Analysis

While care maps enhanced with algorithms can capture all technical data of the care procedures or treatments, O’Neill (2000) further described a method of capturing extra information or **metadata** associated with care map nodes. Metadata is loosely defined as “data about data”; it is extra documentation or data necessary for providing a comprehensive understanding of the information within the nodes (Baca and Swetland, 1998). In this case we can use metadata to capture extra descriptive information about node content beyond simply the text in the nodes. Common examples of care map node metadata include temporal, probabilistic or patient-value data. A care map or algorithm that contains metadata is known as a **decision analysis**.

Some of the care maps encountered in this project use the metadata terms value-added (VA) and non-value-added (NVA) to describe individual nodes’ patient-value information. A VA step in care is one that directly contributes to the patient’s improvement (for example, administering antibiotics to fight the infection). An NVA step, while part of the care process, does not add immediate value to the patient (for example, waiting for blood test results).

Most NVAs encapsulate the key concept of “waiting” for a certain result; when a patient has to “wait” for test results or medication, it is not value-added for the patient. A decision analysis has provision to store this “waiting” time as temporal metadata, that is, information about time taken for an NVA (O’Neill, 2000). Instead of simply flagging nodes as NVA, we can now store time quantitatively (for example, “waiting for blood test results for 24 hours”). This provides an added layer of granularity in the care map, because it is conceivable that two sepsis care maps from two EDs can contain the same NVA step which takes a *different* amount

of time per ED. It would conceivably be extremely useful to examine temporal differences in care processes as part of a group of indicators of care effectiveness and standardization.

1.1.1 Clinical Practice Guidelines - An Evidence Based SCA

SCAs can offer many benefits to both clinicians (nurses, physicians and others) and health administrators to improve care delivery. As health care practice continues to embrace multiple disciplines, SCAs provide a visible flowchart of decisions and serve to minimize clinical decision-making inconsistency. They are excellent records of patient care that can be used as reference material for clinicians and learning material for students to understand complex elements of care in multidisciplinary environments (O'Neill, 1994; Cioffi and Markham, 1997; O'Neill, 2000).

So far, we have seen a hierarchical system of SCAs where the simplest level, the protocol, is encapsulated by care maps, which are further encapsulated and enhanced by algorithms or decision analyses, as shown in **Figure 1.1**. However, evaluating SCAs as they are currently implemented can be challenging. While they may serve as effective training and reference materials, SCAs are qualitative descriptions of the care process. This makes it difficult to smoothly incorporate new and advanced care techniques based on emerging knowledge; it also becomes arduous to include real-time surveillance data from existing pathway applications through the use of modern data capturing technologies. There is an element missing from classical SCAs that would allow them to withstand scientifically rigorous scrutiny and evaluation, that is, the ability to use **evidence** in planning and implementing care, for without evidence, there is no guarantee that the provided care is accurate, efficient and effective (Gaddis et al., 2007). There exists a new class of SCAs that are developed using **evidence-based medicine**, hereafter referred to as Clinical Practice Guidelines (CPGs), which addresses many of the shortcomings of classical SCAs and allows for rigorous evaluation. It is possible to implement evidence-based medicine at any of the SCA levels; for example, Evidence-based Clinical Algorithm (EBCA) were implemented at the algorithm level of the SCA (Gaddis et al., 2007), but the fact remains that any implementation of CPGs is

one that is backed by considerable evidence or goal-directed therapy (Picard et al., 2006).

Rather than implementation at the algorithmic level, CPGs propose a fundamental change at the basic (protocol) level, namely the use of evidence-based medicine in protocol design. Evidence-based medicine is a paradigm where care approaches, diagnoses, treatments and interventions are studied and recorded in great detail, and then this empirical evidence is applied back towards the care practice to make improvements. In other words, the initial evidence is gathered through a thorough literature review, and once assembled into SCA format, prospective data collection during practical application of the SCA leads to the generation of new observations that can further refine and guide prospective evidence-based care. This process can be repeated as many times as possible so as to animate continuous quality improvement to care delivery (Gaddis et al., 2007, 2008) (**Figure 1.3**). By effecting such a fundamental change to the root of the care methodology, CPGs propagate important changes to every level above the protocol, that is, the care map, algorithm and decision analysis levels (**Figure 1.1, Figure 1.3**). It is an attempt to incorporate the latest research evidence in clinical care, and by extension, into care maps; over time, the goal is to conserve VAs and remove or improve upon NVAs. Rather than mapping the clinical journey of a patient by documenting practices alone, the CPG care maps identify the most important VAs and implement a clinical pathway that helps achieve the desired treatment steps.

Figure 1.3 describes the advantage of using evidence-based medicine along with the SCA defined in **Figure 1.1**. Consider the sepsis case study from before, except this time, the protocols are designed by using evidence-based approaches. The care maps and algorithms are implemented on top of the protocols as per the norm, and sepsis care proceeds as prescribed by these CPGs (step 1 in **Figure 1.3**). Now, however, as new information about sepsis care arises through literature, research or breakthrough drug design, care practitioners can go back and revise the protocols to include this new information (steps 2, 3 and 4 in **Figure 1.3**). Once the improved protocol is introduced back into the CPGs (step 5 in **Figure 1.3**), the care maps and algorithms automatically reflect this change as well, and it propagates very quickly into an evolutionary care practice. This cycle can repeat as new data is gathered

using this new care approach, resulting in continuing changes being made at the protocol level. This form of positive feedback through the use of evidence-based medicine allows CPGs to evolve over time (a feature lacking in other SCAs). For example, multidisciplinary sepsis care at Beth Israel Deaconess Medical Centre (BIDMC) was recently revamped through the implementation of a new protocol for nurses, dubbed Multiple Urgent Sepsis Therapies (MUST) (Picard et al., 2006). MUST was implemented successfully and the protocol evolves and is modified as new research and methods are developed for multidisciplinary sepsis treatment (Picard et al., 2006).

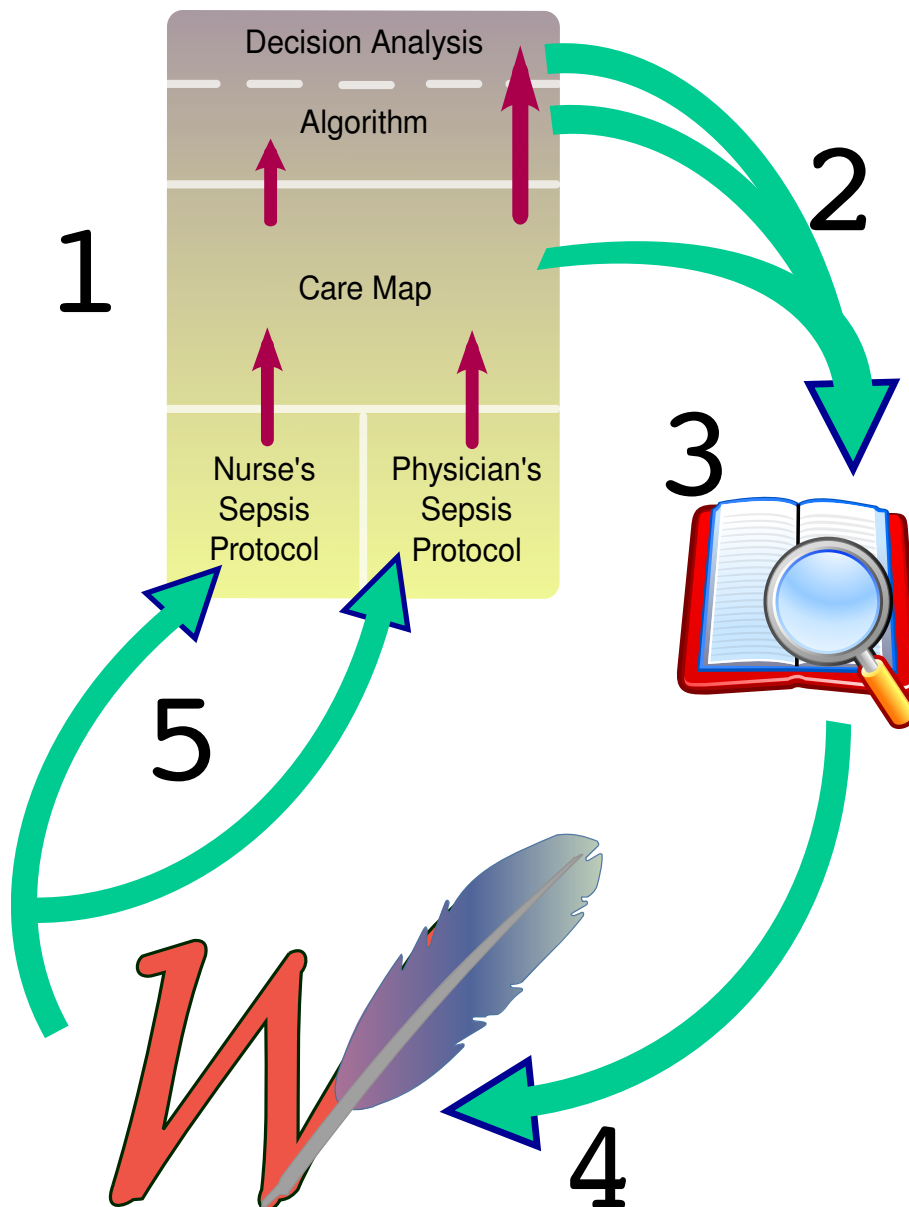


Figure 1.3: The positive feedback cycle generated by the use of evidence-based medicine. 1) The CPG defined in **Figure 1.1**. 2) Every level above the protocol is 3) analyzed for potential improvements which are then 4) inserted into the protocols. 5) The modified protocols are re-implemented into the CPG.

1.1.2 CPGs as an Aid to Electronic Decision Support

CPGs have an added benefit of potentially easy integration into electronic decision support systems, and in the future, Electronic Medical Records (EMRs). Computers, Internet connectivity and a new era of smart phones lend themselves well to access of and data input EMRs. While EMR adoption is not yet widespread, there is considerable evidence for their effectiveness, especially from an evidence-based medical standpoint (McDonald, 1997; de Lusignan et al., 2002; Bates et al., 2003; Wang, 2003). Electronic deployment of CPGs integrated into EMRs can make them available at care giver's fingertips, on demand; it can enable patient-specific care, especially since CPGs allow for flexibility on the care giver's part (Boxwala et al., 2001). Their advantages as far as being reference and teaching materials can be exploited to the fullest when they are available on demand since clinicians can refer to them on a smartphone or students can utilize them in real time as they learn how to treat patients (Boxwala et al., 2001).

Before CPGs can be integrated into electronic decision support, numerous obstacles must be overcome and questions asked and answered. We must now ask ourselves how to incorporate CPGs into electronic decision support systems.

The nature of a CPG brings it tantalizingly close to the era of computational decision making; while the artificial intelligence (AI) required for a computer to making patient-centric decisions via CPGs is still out of reach, computers can still replace manual work in certain areas pertaining to CPGs. One of the motivators for computational analysis of CPGs is the notion of multiple facility implementation analysis. While MUST was successfully implemented at BIDMC, that was still a case of just one care facility using the new protocol (Picard et al., 2006). With CPGs, one of the natural goals is to gain broad adoption across various care facilities (Boxwala et al., 2001). This would ensure that, in a chain of public hospitals, for example, the similar quality of care could be expected regardless of which hospital the patient chooses. However, it is difficult to provide exactly the same implementation of the CPGs across a number of institutions because care facilities vary in size, location, staff and other infrastructure. Furthermore, application of CPGs on patients also varies from one individual to another due to the very nature

of the guidelines themselves; the goal of any SCA, including CPGs, is to guide care givers in administering standardized care while still providing flexibility for the care givers in terms of patient, environmental (care facility) or situational context (Boxwala et al., 2001; Gaddis et al., 2007).

Thus, from an evidence-based medical perspective, it would be extremely helpful to track and compare CPGs implementations, both across multiple care facilities (cross-sectionally), as well as within a single care facility over time (longitudinally) (Picard et al., 2006). As the input data set grows and more and more complex CPG implementations (such as care maps and algorithms) are added to the analysis, their detailed text-based narrative format becomes impossible to compare manually. Thus, computational analysis is necessary to obtain the aforementioned comparisons. If a manual approach were used, it would be impossible for a care giver to examine a guideline or a group of guidelines and pinpoint recommendations for tailoring towards specific patient requirements based on the care facility's infrastructure (Boxwala et al., 2001). Qualitative description of CPGs of different EDs is difficult if not impossible, but by having the ability to convert CPGs into entities suitable for computational analysis, we can not only compare similarities and differences across EDs but also *quantify* the variance. With a computational, quantitative approach, the computer is tasked with examining CPGs implementations and performing the necessary quantitative analyses on the data to provide the care giver with support in making decisions.

CPGs and care maps are normally created, stored and shared in document formats such as Microsoft Word or Portable Document Format (PDF), or image formats such as Joint Photographic Experts' Group (JPEG) or proprietary flowchart formats such as Microsoft Visio. While these formats are electronic and could hypothetically allow guidelines to be retrieved in clinical settings through EMRs, they are not computer-interpretable. While this may not be of concern to the average care-giver due to the ubiquity of these document standards, noncomputable formats do not integrate well into electronic decision support since the semantic knowledge stored within the CPGs cannot be utilized for computational analysis (Boxwala et al., 2001). Thus, while decision support systems can provide the static text-based CPGs for on-demand reference, there is no actual computational decision-making.

That part is left up to the care giver. Also, critically, manual analysis cannot provide quantitative information. If CPGs were compared using computational comparison techniques, we could obtain quantitative data rather than qualitative data alone.

1.1.3 The Importance of Quantitative Information

Quantitative comparisons of CPGs implementations can help answer the following questions:

1. While it is possible to use evidence to guide care, what does it actually mean to improve a CPG through evidence (that is, for the next iterative implementation of the CPG)?
2. How do we evaluate CPGs; how can we be certain that we are in fact providing better care than the previous care iteration?
3. Finally, how do we use these evaluation methods to ensure that CPGs are implemented similarly across multiple care facilities?

There is an advantage to being able to ascribe numerical values to CPGs through quantitative measurement. By assigning numbers, we gain a whole new dimension of data. Rather than saying, for instance, that “The sepsis CPG implementation at Hospital A is better than the one at Hospital B,” we can qualify the statement by providing relative measurement: “The sepsis CPG implementation at Hospital A provides care, on average, ten minutes faster than Hospital B. This translates to a 5% increase in patient outcome.” While the numbers in this example are completely contrived, the second statement provides tangible and measurable examples of improvement and is evidence based as opposed to the first one; it is open to further analysis about how Hospital B can improve its patient outcome. It is also evident here that, as the input set of CPG implementations increases (or in the example, the number of hospitals increases), only a computational approach (as opposed to manual) can sift through the data and carry multivariate analysis between different care facilities simultaneously.

1.2 Making CPGs Viable for Computational Analysis - Project Objective and Approach

This project aimed to make CPGs electronic and analyzable through computational methods. It involved two parts:

1. Research and development of a sharable document standard for CPGs by building upon an existing document standard for storing biological pathway information.
2. Analysis of CPGs through development of a special decision support software system that incorporates biological pathway alignment approaches from the disciplines of Bioinformatics and Systems Biology.

1.2.1 Types of Analyses

We can safely assume that improvement of CPGs over multiple iterations can be studied by examining patient benefits or outcomes; the more patients that can be treated successfully in as efficient a manner as possible, the better the CPG implementation. There is considerable research and evidence available to show that any form of SCA can improve patient outcome (Van den Berg and Visinski, 1992; Cole et al., 1996; Venner and Seelbinder, 1996; James et al., 1997; O'Neill, 2000; de Lusignan et al., 2002; Picard et al., 2006; Gaddis et al., 2008), so examining outcome numbers should provide a good measure of CPG effectiveness. Computational comparison of CPGs for a certain procedure or intervention gives us three important pieces of data, as summarized in **Table 1.1**.

Cross-sectional analysis across multiple care facilities provides us with information on how these facilities use their resources and infrastructure to implement a set of CPGs. Comparisons can reveal, for example, why one ED's implementation contributes to overall better patient outcome than another's. The facilities with non-optimal results can then examine the more effective implementation methods

Analysis Method	Data Obtained	Number of care facilities
Cross-sectional	Similarities and differences	Multiple
Longitudinal	Similarities and differences	One
Consensus	A set of conserved CPG steps	One/Multiple

Table 1.1: The types of data obtainable from computational analysis of CPGs for a given procedure or intervention. Each analysis method provides specific types of data based on the number of care facilities considered in the analysis.

to find ways of optimizing their care within their infrastructure limits. In a long-term research study of CPGs implementations, multiple such revisions are possible, which fully leverage and contribute to evidence-based care.

In **longitudinal analysis**, the emphasis is placed upon how a single care facility is able to change its care approach over time in order to improve patient outcome. This is again reflective of positive feedback in evidence-based care, since measurements are taken at discrete time intervals and recommendations to change the CPG implementation are prescribed.

Consensus analysis describes a case where all CPG implementations are computed and the set of converging CPG steps is the desired output. Since implementations differ across care facilities due to institution-specific nuances (Boxwala et al., 2001; Gaddis et al., 2007), it is useful to know which steps are implemented equally. This gives care givers an idea of the common core modules and steps in the CPG being studied, and thus, provides evidence about the static and dynamically-changing portions of CPG implementations.

1.2.2 Considerations in Building the CPG Document Standard

In order for CPGs to be eligible for computational analysis, a special document standard needs to be developed. In order to develop this document standard, it is imperative to understand how CPGs are currently developed and shared. Evidence-based medical research is applied to an identified clinical problem in order to create

recommendations in narrative form (the CPGs themselves). These CPGs are then subjected to approval by a consensus committee, usually the group of care givers responsible for developing and spreading the CPGs. Once the guidelines are sent out to participating care institutions (for that is one of the goals of CPGs, to be sharable across multiple facilities), they are implemented in an institution-specific manner, which is where the individual implementation changes happen (Boxwala et al., 2001; Picard et al., 2006). Sharing the CPGs across institutions in care map format is more concise and visually intelligible compared to the CPG narrative format. Care maps are represented graphically, in flowchart form, with a rigid, predetermined set of symbols in order to be understood universally (**Figure 1.2**). Before continuing however, it is worthwhile to define some care map structural terminology.

A Simplified Care Map

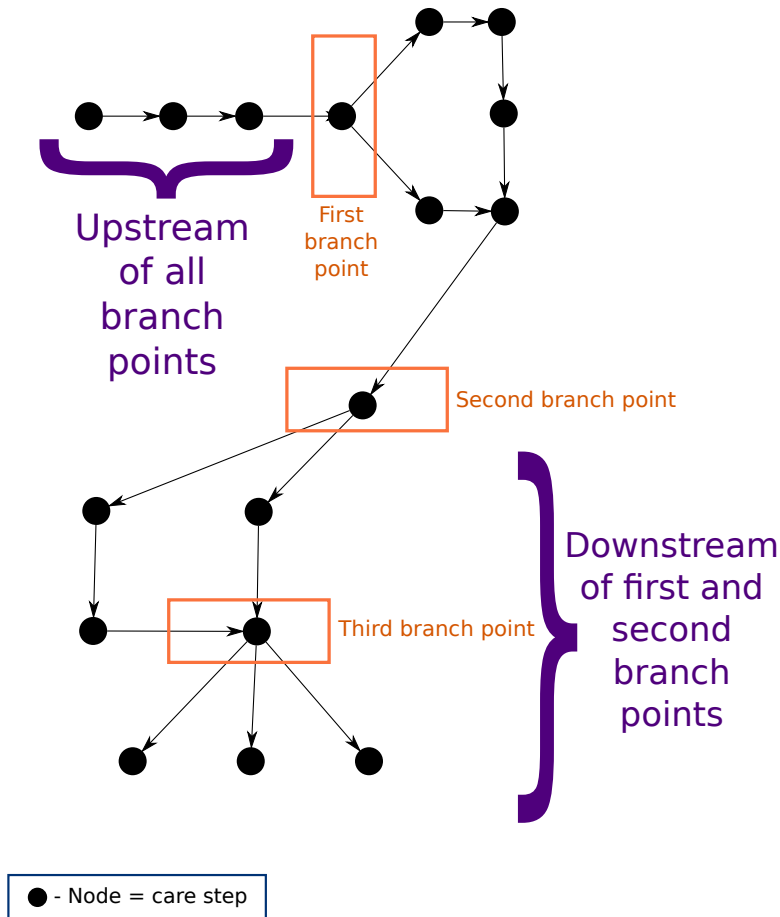


Figure 1.4: The basic components of a care map showing branch points and the definitions of “upstream” and “downstream.”

Figure 1.4 shows a simplified care map structure without any textual content, in order to emphasize the common terminology used to address care map structure. A care map contains nodes (circles in **Figure 1.4**); each node contains a quantized, actionable step derived from the CPG which is the basis of the care map (not shown in **Figure 1.4** for simplicity). Nodes are connected by arrows; some nodes are branch points, which lead down two separate paths. The branch point is

often a decision-making or algorithmic step, and sometimes, two or more branches may collapse back together into one. When describing care map steps before or after a branch point, it is useful to use the terms “upstream” and “downstream”, respectively. Thus, by referring to a care map, care givers can visualize the care steps involved in that particular procedure or intervention. In subsequent text, “care map” and “CPG” are used interchangeably, as care maps are visual representations of CPGs.

In this project, we work with care map representations of CPGs when considering the features that our document standard should provide. Boxwala et al. (2001) neatly describe the three important considerations in developing a sharable, computable CPGs format: institution-specific tailoring and modifications, scope, and sharing method. Each consideration can affect the sharability, computability, or both aspects of the care map.

1.2.2.1 Institution-specific Customization of Care Maps

The document standard should allow each institution to create care maps that can fully capture the workflow of that particular institution. Institutions should also be able to modify their care maps over time; a static document format such as a JPEG image does not allow easy modification and updating (Boxwala et al., 2001). The document standard should allow institutions to create and use a standard set of vocabulary for describing individual nodes in their care processes. This makes computational analysis much easier, as described in **Section 2.2.1.2**.

1.2.2.2 Scope of a Care Map

Defining the scope of the text contained within a node is very important when creating and describing care map node data. For example, in the case of a sepsis CPG care map, one ED may create three adjacent nodes that say “Take patient’s registration details,” “Locate bed for patient” and “Place patient in bed.” A second ED, working with the same sepsis CPG may define a single node that says “Register

patient and place patient in bed.” The scope of each care map is now different. The second care map assumes that, by placing a patient in a bed, the bed has already been located, while the first care map isolates the location of the bed into a separate step. While this may not affect the sharability of the care map document itself, it does affect computability, since the computer now needs to explicitly know about the scope of each care map. The two care maps above are essentially describing the same series of steps, but having three nodes instead of one affects the content of the care map nodes greatly. A computer cannot make sense of the two care maps unless the scope is kept consistent (Bell et al., 1991; James et al., 1997; Advani et al., 1998; Bernstam et al., 2000; Boxwala et al., 2001).

1.2.2.3 Care Map Sharing and Delivery Methods

The sharing, or delivery method is the actual transport and transfer of care maps across multiple institutions. Here is where we encounter the previously described problem about multiple document formats. This is both a sharability and computability issue, since each institution must be able to read the document format used by another. All too commonly, the document formats used today are frequently non-computable, and so this is a large motivation to develop a proper CPGs or care map document standard. By creating a standard for care map sharing, we can also specify a method of care map delivery, be it through email, web-based or other methods (Boxwala et al., 2001). This concept is further discussed in **Section 3.2.1**.

1.3 Considerations for Computational CPG Analysis

As stated before, by considering the care map representations of CPGs, we can potentially build a robust document standard to capture the information in a care map. However, the second purpose of this project, namely, the analysis of CPGs, should also be achievable through the use of the same document standard. Otherwise the standard does not satisfy all our requirements. The idea here is that care maps,

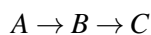
when described in our custom document standard, can be fed into a software-based analysis engine that understands the information in the document standard since the use of the document standard ensures that the care maps are now in a computable format. In this project, we required three types of analysis data as outlined in **Table 1.1**.

In order to satisfy both goals of the project (CPG document standard development and care map analysis), a translational approach was developed using Bioinformatics concepts.

1.4 Adaptation of Biological Pathway Alignment Concepts

Bioinformatics is the application of Computer Science to biological problems, mostly of the genetic or molecular variety. Research problems in Bioinformatics can range from high-level Graphical User Interface (GUI) design for a protein visualization tool to highly complex mathematical proofs and computational algorithms to crunch the massive amounts of data generated by high-throughput genomics experiments (Pinter et al., 2005). Two research areas within Bioinformatics are relevant in this project: development of document standards for sharing biological or molecular pathway data (a research question shared with the closely related field of Systems Biology), and computer algorithms for analysis of these pathways (not to be confused with SCA algorithms!).

Biological and metabolic pathways (hereafter used interchangeably) can be represented graphically as flowcharts where nodes are substrates or biochemical species, and arrows between the nodes are biochemical reactions (Chen and Hofstaedt, 2005). The simplest pathway is composed of two nodes (describing two biochemical species), connected by a single arrow. This looks exactly like a simple chemical reaction, as seen in **Definition 1** below.



Definition 1. A simple chemical equation with two reactions. In the first reaction, reactant *A* is converted to product *B*. In the second reaction, *B*, now a reactant, is then converted to product *C*.

It should be straightforward to note that, on a high level, pathways resemble care maps, visually. The reactants and products are akin to care map nodes, and the arrows serve the same purpose in both cases, to show the direction of process workflow. Thus, if there exists a document standard for describing biological pathways, then we might be able to leverage it in creating the care map document standard. Indeed, there are multiple document formats for describing and sharing biological pathways. This project evaluated the use of one, Systems Biology Markup Language (SBML), as a potential framework for developing the care map document standard (Strömbäck and Lambrix, 2005; Gauges et al., 2006; Gillespie et al., 2006; Bornstein et al., 2008).

Regarding analysis of care maps, if they can be expressed as biological pathway analogues, there are a host of **alignment approaches** defined by Bioinformatics for the study of pathways. The word “alignment” describes the act of computationally “stacking” biological pathways on top of each other. The algorithm then notes similarities and differences by essentially “tracing” the common paths present in the care maps. Such alignment approaches can potentially give us the cross-sectional, longitudinal and consensus data we are looking for, and the algorithm used in this project, called IsoRank, appeared to fit the bill perfectly (Singh et al., 2008). While an implementation of IsoRank has been produced by Singh et al. (2008), it suffers from some technical drawbacks; a customized version of IsoRank was rewritten in this project (**Section 2.3.4**).

1.5 Context of this Project

The concept of this study – the computational analysis of CPGs – owes its genesis from a provincial collaborative initiative amongst the EDs which share the common vision of improving the delivery of emergency care in the province of British

Columbia. Under this grass-root initiative, called Evidence to Excellence (E2E), physicians, nurses, and administrators from EDs situated in small and large communities have committed to work and learn together to improve and harmonize clinical care to ensure optimal patient outcome. E2E was successful in obtaining competitive funding from the Canadian Institutes of Health Research (CIHR) in 2009 to harmonize on the ED management of sepsis (severe infections) (Ho et al., 2009). As part of this proposed project, EDs would actively work together to implement sepsis CPGs in their respective organization. This would present a great opportunity for collaboration and mutual learning of CPGs implementation in three ways:

- Similarities and differences in CPG implementation in different settings, from urban to rural locations, and from different composition of personnel in the EDs.
- The longitudinal evolution of CPG implementation in one ED over time, and how much other EDs' implementation strategies influence this ED's CPG implementation approach.
- The degree of variation of CPG implementation in the province's different EDs in the beginning and at the end of the project, and see how much harmonization has taken place over this period.

Having a quantifiable way to compare CPG implementations as care maps would conceivably help in these three types of comparisons. Currently, the larger project is collecting baseline data for the pivotal CPG steps in sepsis care. The goal is to quantify similarities and differences as described in **Section 1.1.3** in order to continually add to and improve the sepsis CPGs themselves; the baseline data will serve as the reference when comparing CPGs implementations through the computational analysis approach of this project. The methods presented here encapsulate step 3 of **Figure 1.3**, that is, the comparison of CPG implementations against the baseline collected in the larger study. The hope is that the computational analysis of CPG implementations presented in this project will aid the larger study in im-

proving CPGs by illuminating cross-sectional, longitudinal and consensus nuances between EDs.

1.6 Project Goals and Anticipated Outcome

The goals of this project were threefold:

1. **Translation** - To use an existing Bioinformatics document standard and adapt it to the case of describing, creating, modifying and visualizing sepsis care maps.
2. **Analysis** - To use an existing Bioinformatics pathway alignment approach and adapt it to alignment of the sepsis care maps.
3. **Implementation** - To develop a custom software system that performs the two tasks above in a unified manner, making it easy for care providers to transition to this new care map creation and analysis approach.

Chapter 2

Methods

The principal methodology revolved around development of a software system, codenamed “Chequers,” that incorporated all features required to create care maps based on SCAs using a customized document standard, and usage of the IsoRank algorithm in aligning care maps. For this purpose, we used two sepsis care maps obtained from EDs at two hospitals in British Columbia, Canada.

2.1 Translation - Choosing SBML to Design the Care Map Standard

SBML is an Extensible Markup Language (XML) derivative that is used in Systems Biology and Bioinformatics to represent biological pathways. The benefit of using XML is that one can define one’s own schema, or structure, of the data stored within the document. This allows for easy development of new document standards. SBML was chosen because it already had a defined structure to store biological pathway data which allowed for easy translation of care maps into pathway analogues. It can also be easily incorporated into any programming language (Strömbäck and Lambrix, 2005; Strömbäck, 2006).

Recall the three major considerations in developing a document standard: institution-specific requirements, scope, and method of sharing. Ideally, the document standard should accommodate these considerations while providing computational viability. Boxwala et al. (2001) describe the major requirements for a sharable, computable CPG format, which were adapted in the development of the guideline document standard in this project. SBML was chosen as the base format because it satisfied the maximum number of requirements. The requirements pertinent to the implementation of the care map standard are discussed below.

Firstly, the standard should allow for representation of **different types of guidelines** (Boxwala et al., 2001). In this project, we focused on sepsis CPGs, but there is no reason why the document standard should not be able to accommodate other procedures or interventions. By recording the care map structure and content in our document standard instead of the narrative form of a CPG, we can ensure broad compatibility. SBML is designed for recording any form of biological pathway (Strömbäck and Lambrix, 2005), so it fits our requirement here.

Secondly, the standard should be able to accommodate **modifications by local institutions** (Boxwala et al., 2001). There should be some sort of software interface to allow institutions to create and modify care maps using the new document standard. In this project, a custom care map creation and analysis software suite was developed to tackle this problem. SBML provides an Application Programming Interface (API), `libsbml`, for most popular programming languages such as C/C++ or Python, and so it is easy to integrate SBML functionality into custom software such as the one developed for this project (Bornstein et al., 2008).

Thirdly, the standard should be able to support **revision control** (Boxwala et al., 2001). Revision control systems, such as Subversion which was used in this project to maintain the software code base, maintain a history of changes made to documents. At any point, it is possible to go back to a previous revision or merge changes made by multiple users at the same time. This avoids having to email files back and forth to people. A long thread of emails with the same file can often be confusing, as one sometimes forgets which is the latest version. By using revision control, these problems are avoided. Revision control also works best with plain

text files, rather than images or proprietary document formats such as Microsoft Visio (Pilato et al., 2008). Since our care maps are described in a text-based format (SBML) and are intended to be used and modified in a collaborative manner, they are conducive to revision control.

Thus, SBML provides a near-perfect drop-in replacement for current non-computable care map formats, and was chosen as the paradigm for building the care map document standard.

2.2 Analysis - Pathway Alignment Approach

Just as a drop-in replacement was found in SBML for care map document standard development, the goal for alignment was similar. Since we managed to translate care maps into biological pathway analogues, it made sense to use an existing biological pathway alignment technique in order to compare the care maps. We found one algorithm, IsoRank, that could do what we hoped (Singh et al., 2008). Furthermore, there was an IsoRank executable program available, so rather than developing an implementation of the algorithm, an attempt was made to utilize the existing software.

2.2.1 Notes on the IsoRank Pathway Alignment Algorithm

In Bioinformatics, there are two kinds of alignments: global and local. Alignments, used on genetic data such as protein or DNA sequences, provide a measure of the degree of similarity or difference between two or more sequences. Global alignments consider the entire length of the sequence, while local alignments look for sections of highly matching sections (Needleman and Wunsch, 1970; Smith, 1981). IsoRank is a global pathway alignment algorithm. It uses two important pieces of information in order to generate alignments between two or more biological pathways: 1) interaction data and 2) similarity data.

2.2.1.1 Interaction Data

Interactions describe the overall structure of the pathway, that is, how the nodes are connected to each other (**Definition 1**). IsoRank takes into account these similarities when generating alignments (Singh et al., 2008). Since our input data consisted of care maps, there needed to be a way to automatically extract interaction and similarity data from the care maps in order to feed IsoRank with the necessary inputs.

The interaction data for care maps is relatively easy to compute. It is essentially a breakdown the care map into discrete “reactions,” where each reaction contains a “reactant” and a “product” (**Definition 1**). Conveniently, SBML uses similar reaction notation to store its pathways, so obtaining the reaction data from the input SBML care maps was straightforward. The details of how this was done in software are discussed in the section on software design and development.

2.2.1.2 Similarity Data and Vocabulary Control

Since IsoRank’s usual input pathways are composed of protein sequences, it is fathomable that protein analogues exist in different pathways. IsoRank’s required similarity data is obtained by comparing protein sequences in one pathway with protein sequences in another. IsoRank asks for Basic Local Alignment Search Tool (BLAST) scores between all the proteins as similarity data input. BLAST is a sequence alignment algorithm commonly used to identify homologous amino acid sequences. BLAST assigns a score of 0 for two perfectly matching sequences, that is, two identical proteins. For scores that are greater than 0, we can derive a spectrum of protein homologues; scores closer to 0 can indicate homologous proteins (Altschul et al., 1990).

In our use case, we do not have amino acid sequences and so generating the similarity data poses a bigger problem. It is easy to talk about similarities between two proteins by comparing their amino acid sequences, but in care maps we deal with sentences that describe actionable steps. The document standard should give

institutions the flexibility of describing their CPGs in their own words, yet globally these words may carry similar meanings as they apply to other institutions. We therefore need to derive a similarity measure from these **semantic differences** between the words and their meanings Boxwala et al. (2001). Computationally however, how do we quantitatively compare sentences by their similarities? For example, the sentences “Nurse sees patient” and “Patient is seen by nurse” have similar meaning to us, but to a computer they are completely different. The use of varied vocabulary to describe the same essential task or care step is detrimental to computational analysis because a computer would regard the two sentences as being different (Boxwala et al., 2001). We had to devise a way of accommodating semantic differences while still generating similarity data for IsoRank.

One solution is for the software system to be able to compute sentence similarities automatically. Bioinformatics provides an answer here. Our care maps are stored in SBML format, which is a derivative of XML. XML is used for other related schemas (highly structured documents), including a particular class of documents called **ontologies**. Ontologies, such as Web Ontology Language (OWL) or the popular Gene Ontology (GO), are used in Bioinformatics to describe semantic relationships between words and phrases (Strömbäck, 2006), and since both OWL and SBML are XML document standards, it may be possible in the future to combine the two in order to take care of the semantics issue. Ontologies are often packaged together with a **reasoner** that does the work of comparing input vocabulary to its existing semantic database to find synonymous terms (Cary et al., 2005; Str, 2006; Strömbäck, 2006; Wächter et al., 2006; Lambrix et al., 2007). The idea here is that, by including the ontology definition inside the SBML document, we could conduct a pre-processing step where we feed the SBML document into the reasoner and obtain similarity data.

In this project however, we needed a solution for faster implementation and so we employed another Bioinformatics concept derived from the principle of ontology described above.

A **controlled vocabulary** is similar to an ontology, so similar in fact that it is often defined as a simple ontology. It contains a specific vocabulary database that must

be adhered to when describing a concept (Lambrix et al., 2007). By controlling the vocabulary used in care map creation, we can enforce a standard across all input care maps (thus, every care map will say “Bedside nurse sees patient” as per the example above). While this method is restrictive, it adheres to Boxwala et al.’s rule that the document standard be open to institution-specific nuances and modifications. Our SBML document itself poses no restrictions on the language and vocabulary used. We chose instead to control the vocabulary at the user interface level by allowing the user to pick from a standard vocabulary and create new terms only when absolutely required (discussed in **Section 2.3.3**).

Now, by using a controlled vocabulary when creating care maps, we had a way of providing IsoRank with the similarity data it needed. In fact, it became a binary score case, where two sentences are either evaluated as being equal or not. Some of the controlled vocabulary we chose is listed in **Figure 2.1**.

- ```

1. Patient waits in waiting room
2. Patient goes to bed
3. Patient sees Triage Nurse (TN)
4. Patient sees Bedside Nurse (BN)
5. MD orders tests, IV fluids, diagnostics
6. Start IV
7. Disposition
8. Call BC Bedline
9. Patient discharged

```

**Figure 2.1:** Some of our controlled vocabulary.

With this set of controlled vocabulary, the input data was ready for utilization by the IsoRank software. The IsoRank implementation developed by Singh et al. (2008) to accompany their publication was a GNU/Linux-only application with a command line interface, however, we were unable to make the implementation work with our data. While IsoRank’s algorithm is public-domain, we could not obtain

the source code in order for us to contribute to fixing the software. As a result, we chose to spend the last few weeks of the software engineering phase carefully analyzing IsoRank’s algorithm and writing code to reproduce its functionality.

Although IsoRank is capable of performing multiple alignments - the ability to analyze more than two pathways at a time, given the unexpected time we needed to re-implement the IsoRank software, we limited our current test case on a pairwise alignment comparison only - analyzing two pathways at a time. Multiple alignments are much more complicated to implement, but by proving that the pairwise alignment approach worked, we could build a foundation for this project which would then allow us to extend the IsoRank module to the multiple alignment case at a later date.

## 2.3 Implementation - Software Design and Development

Having defined a care map standard and an alignment approach, it was now necessary to build a software system that incorporated these concepts. Software design is a critical component of the software development pipeline. It was also imperative that the software adhere to principles and ethics governing the Free Software movement (a stricter sibling to the Open Source Initiative; all Free and Open Source Software (FOSS) is Open Source Software (OSS), but the reverse is not necessarily true), as this ensured the software was built free of licensing and other proprietary requirements (Cass, 2002). Therefore, the software was developed using the free GNU Compiler Collection (gcc) compiler and written in the C++ programming language. The software was named “Chequers” after the author enjoyed a particularly satisfying meal at a pub of the same name located in Oxford, UK.

We had two big criteria in mind when building “Chequers”: **cross-platform compatibility** and **modularity**.

### 2.3.1 Cross-Platform Compatibility

Cross-platform compatibility means that the software will function in the same fashion regardless of the end-user's operating system, be it GNU/Linux, FreeBSD, MacOSX, Microsoft Windows, Google Android or the Symbian cell phone operating system. There is an extensive suite of FOSS libraries for C++ known as wxWidgets, which allows for building cross-platform software and GUIs in an easy, standardized manner Smart et al. 2005. wxWidgets was used to build the user interface and functionality for Chequers.

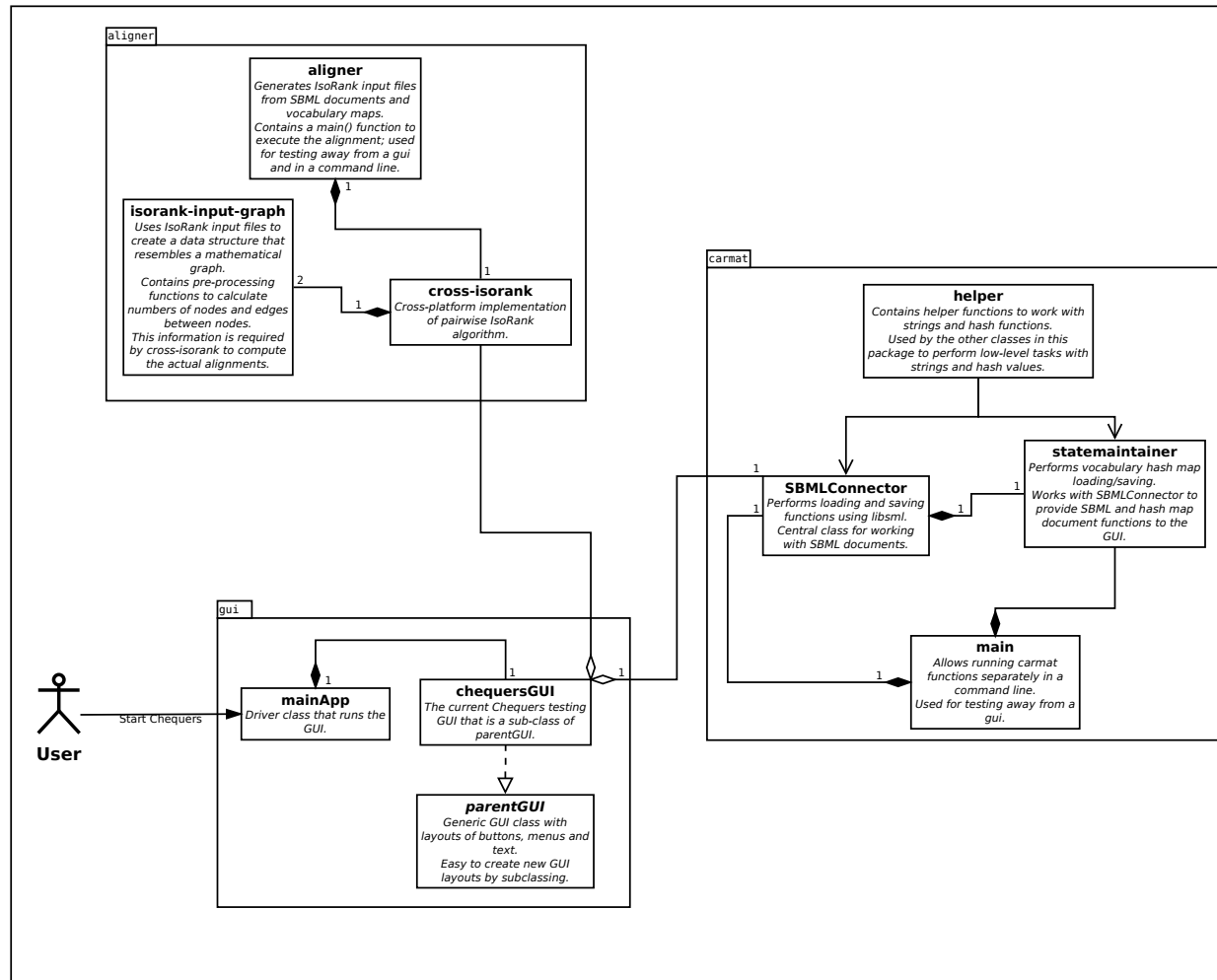
### 2.3.2 Modularity

Modularity is a concept in software design that strives to decompose key functional elements of the software into separate "modules." This allows changes to be made to one module without affecting another, and makes software maintenance much easier (Parnas, 1972). Modularity was especially important to us in the case of using IsoRank. IsoRank is one particular algorithm and there may be a better performing algorithm in the future. By separating all of IsoRank's functionality into a separate module, we can easily replace it with another algorithm without affecting any other part of the software system.

As Chequers currently stands, its modules are:

1. `carmat` - Care map translation into the SBML-derived standard, including controlled vocabulary functionality.
2. `aligner` - The IsoRank implementation used to align care maps.
3. `gui` - The GUI.

**Figure 2.2** shows how Chequers' modules are organized. It also shows the individual classes, or objects, that make up each module, and describes the interaction between the various module components. When the user starts the Chequers application, the `mainApp` class is invoked; this leads to series of invocations based on how the user is interacting with the GUI.



**Figure 2.2:** A class diagram showing Chequers' three main modules and classes that make up each module. The `chequersGUI` class integrates functionality from `aligner` and `carmat` and presents it to the user in graphical format.

## 2.3.3 The `carimat` Module and Manual Care Map Curation

### 2.3.3.1 `carimat` Module Design

SBML has a set of C++ libraries, called `libsbml`, which were used in building `carimat` (Gauges et al., 2006; Bornstein et al., 2008). The user can create a care map from scratch using the `carimat` module.

**Figure 2.2** shows that `carimat` contains four classes. The `SBMLConnector` class contains the major functionality to interact with `libsbml`. It allows creation and modification of SBML documents. `SBMLConnector` is **composed** of a class known as `statemaintainer`; the black diamond arrow in **Figure 2.2** indicates composition, meaning that `statemaintainer` is used within `SBMLConnector` to perform its functions, rather than being evoked separately (Gamma et al., 1994) (the significance of the digits on the black diamond arrows is discussed in **Section 2.3.4**). The `statemaintainer` is used to handle vocabulary map functions described below in **Section 2.3.3.2**. The `helper` class provides mathematical functions to work with C++ strings (text) and hash values for the vocabulary map (**Section 2.3.3.2**).

Since current care maps are not in the SBML format, manual translation of current care maps was required. What follows is the general use case for creating a care map through the GUI, which uses the `carimat` module:

1. User chooses 'New' from the GUI. Internally this initializes a new SBML document and a **vocabulary map** (described below in **Section 2.3.3.2**).
2. User adds vocabulary, one at a time. Each vocabulary item is a sentence taken directly from the care map being translated. This is the controlled vocabulary to be used in step 3.
3. Once all the vocabulary is added, the user creates "reactions." Each "reaction" is composed of a "reactant" vocabulary term and a "product" vocabulary term. If we were trying to digitize the example in **Definition 1**, we



would define three vocabulary items: *A*, *B* and *C*. We would then create two reactions: *A* to *B* and *B* to *C*.

4. User saves the care map when all reactions are entered into `carmat`. This triggers the series of events described below.

When the care map is saved, a few things happen. The `carmat` module asks the user for a file name. The SBML document gets populated with all the vocabulary and reactions; the output file is then written, named with the file name chosen by the user. The vocabulary map is also written, using the same file name and the extension “.vocab”. The vocabulary map and SBML document are then archived together in a Tape Archive (TAR) file. A JPEG and Postscript document (PS) image of the care map are also created. This last portion required some assistance from external software.

One way to check whether the SBML care map is equivalent to the starting care map is to visualize it as a flowchart, just as the starting care map is expressed. Unfortunately, an SBML document is anything but a visually pleasing flowchart. Luckily, the people who maintain SBML also released a biological pathway layout and render tool, which automatically generates graphical layout information based on the reactions in SBML document and draws an image in flowchart format (Hucka et al., 2003). Thus, `carmat` calls upon this external layout tool in order to render the care maps as images. The layout tool faithfully renders the information in the care map and produces a human-readable image. This step was important as it ensured that the original and new care map held the same data, and that the document standard still allowed backwards-compatible visual representation through classical image formats.

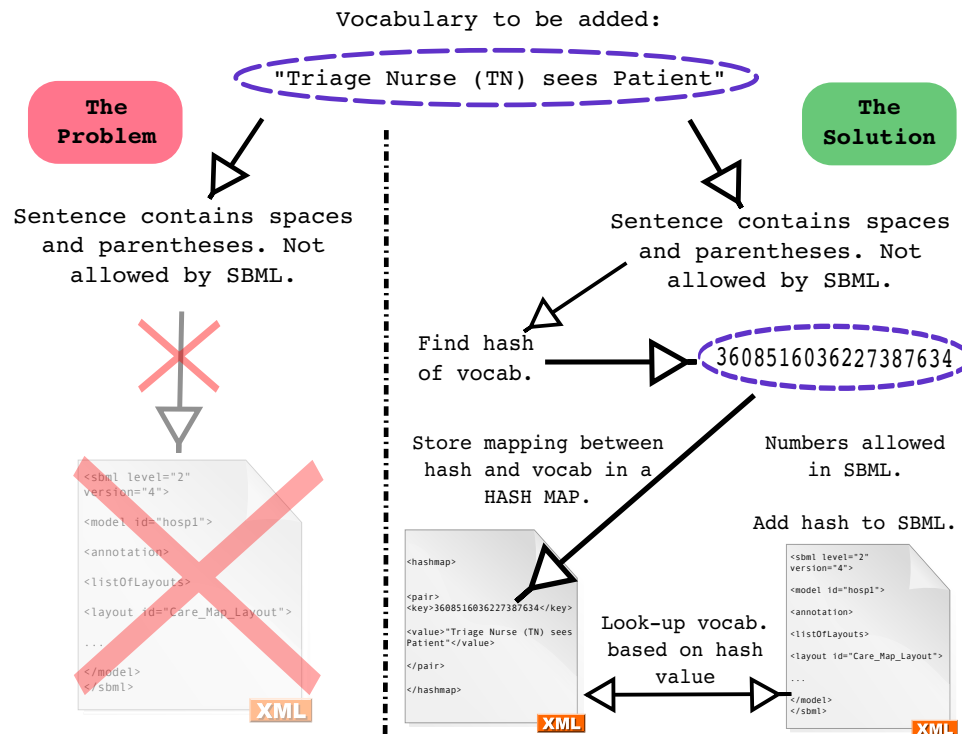
### **2.3.3.2 The Vocabulary Map**

Reactants and products in SBML are tagged with unique identifiers. Since SBML was designed to store protein identifiers in reactions, it enforces a strict convention on identifier naming; simply using the vocabulary itself as the identifier will not

work. Protein identifiers follow a naming convention: a few letters followed by numbers, perhaps segregated by an underscore (“\_”) character. SBML does not allow spaces, which are ever present in our care map vocabulary. Thus, there needed to be a way to transform our vocabulary into SBML-valid identifiers on the fly while still allowing us to trace the identifier back to the original vocabulary.

**Figure 2.3** shows the problem and our devised solution, which involved the use of a **hash map**, a data structure in C++ that stores data as a key/value pair. The key is a unique numeric identifier, known as a **hash number**, which maps to a value. In our case, the values are the vocabulary sentences, stored as C++ `strings`. The keys (hash numbers) are generated using an inbuilt reversible hash function applied to the values (vocabulary sentences). The keys satisfy SBML constraints and can be inserted into the SBML document. Whenever the vocabulary is required, the keys can be used to look up the corresponding vocabulary values in the vocabulary map. Hash map functionality is implemented in the `statemaintainer` class of the `carpat` module (**Figure 2.2**).

This method of mapping by hash numbers serves an important secondary purpose. In the unlikely event that the vocabulary map is ever lost, it is possible to recreate the vocabulary from the SBML document alone. By isolating the hash numbers in the SBML and running the hash function in reverse on them, we can retrieve the vocabulary. This promotes robustness and redundancy.



**Figure 2.3:** The use of a hash map as an intermediary between the vocabulary and the SBML document satisfies SBML's stringent requirements for formatting of the data in the document. Since the hash value is generated from the vocabulary, applying the hash function in reverse on the hash number will return the vocabulary sentence.

### 2.3.3.3 Manual Care Map Vocabulary Curation

One of the tests of the controlled vocabulary was to see if it could be applied with relative ease to multiple care maps. First, a sepsis care map for Hospital 1 was analyzed and the most important vocabulary terms were defined. Some of these terms are reproduced in **Figure 2.1**. These terms give us a place to anchor the controlled vocabulary, and individual EDs would have the flexibility to add other terms as required to describe their process.

This now required manual translation on the author's part, that is, rendering the care map vocabulary as faithfully as possible while trying to use the controlled vocabulary as much as possible. After the first care map was complete, a second care map from Hospital 2 was translated in similar fashion, using the controlled vocabulary generated for Hospital 1 and adding or modifying nodes as necessary. Translation of the first (Hospital 1's) care map took about five hours; most were spent learning and understanding the steps in the care maps, while the last hour involved physically typing out the care map into Chequers for conversion to SBML.

### 2.3.4 The `aligner` module - A Cross-Platform IsoRank Implementation

There was an implementation of IsoRank available, built by the authors of the algorithm (Singh et al., 2008). The original idea was to call the external implementation directly, just as was done with the layout tool in `carpat`. However, the IsoRank implementation required a unique set of input information for the interaction and similarity data that it needed to compute the alignments. The `aligner` class contains a parser that examines the SBML care map document and produces the necessary input files (**Figure 2.2**).

IsoRank requires an interaction file for each care map. The interaction file is basically the same as the list of reactions in the SBML document, so it was straightforward to retrieve all the reactions from the SBML document and process them into interaction files as required.

The similarity files required by IsoRank are files that contain BLAST scores between every “species”, that is, data in the nodes. Here is where a little creativity was required in order to produce a quick and dirty solution. Recall that the notion of “similarity” between sentences cannot be described without the use of an ontology or some other form of semantic data structure. However, since the `caromat` module was already in essence enforcing a controlled vocabulary, this property was leveraged through a modified BLAST algorithm.

BLAST examines nucleic or amino acid sequences and assigns a score between 0 and 1, where 0 implies a perfect match and 1 signifies that the sequences are completely dissimilar. There is a continuum of scores since two amino acid sequences, for instance, could be somewhat closely related without being a perfect match (Altschul et al., 1990). In our case, the “sequences” are actually sentences and we cannot achieve a continuum of scores by using BLAST techniques. Therefore, a simple `QuickScore` algorithm was written as part of `aligner` (in the `cross-isorank` class, shown in **Figure 2.2**), which as it sounds, is a quick and dirty alternative that does not actually use the BLAST algorithm at all. If two sentences (represented as `string` objects in C++) are equal, `QuickScore` assigns a score of 0; if the sentences are not equal, a score of 1 is assigned. Since `caromat` enforces a controlled vocabulary, the user must pick from the vocabulary created in the interaction with the `caromat` module, and so this binary scoring system works well.

At this point, `aligner` was supposed to call the external program, Singh et al.’s IsoRank implementation, but unfortunately there were technical issues in using the implementation and source code was unavailable. This meant that, in order to use the existing `aligner` code which was designed to work with this existing software, a new IsoRank implementation had to be written within the `aligner` module that mimics the implementation created by Singh et al. (2008). It takes in the same input data and generates the same output, and has the added benefit of being cross-platform and free of proprietary licensing restrictions; the implementation by Singh et al. (2008) works on GNU/Linux operating systems only. The added benefit here is that we would be able to ship Chequers without bundling Singh et al.’s IsoRank as well, which avoids any future licensing hiccups that might have arisen.

Our cross-platform IsoRank implementation is able to conduct pairwise (two care map) alignments, as this is the base case for multiple alignments; the smallest number of alignable elements is two. Once we can get reliable pairwise alignments, it is merely a matter of adding in the extra code to allow for multiple alignments. In order to write our version of IsoRank, a free C++ library called `Armadillo` was used for the linear algebraic components of the algorithm. The IsoRank algorithm is implemented within the `cross-isorank` class of the `aligner` module, as shown in **Figure 2.2**.

**Figure 2.2** shows that `cross-isorank` is composed of `isorank-input-graph`. This class takes the input files generated by the `aligner` class and stores them in memory using computational data structures. This speeds up the execution of the application and also provides `cross-isorank` with information it needs to compute the alignments, namely the number of nodes and edges in each care map. The black diamond composition arrow contains the number 2 next to the graph class, and the number 1 next to `cross-isorank`. This means that each time `cross-isorank` is called, it will need two `isorank-input-graph` classes to work. This makes sense because we are attempting a pairwise alignment of two care maps; each care map will need its own `isorank-input-graph` class.

### 2.3.5 The Chequers GUI

The user interface was written completely using the `wxWidgets` API for C++. It supports all the functionality for interacting with the `carimat` module, but does not as yet support `aligner` fully since this module has just gone through the testing stage. The `aligner` module is, however, accessible from a UNIX-style command line (by invoking the `cross-isorank` and `aligner` classes from the command line), and the GUI does have some functional placeholders for incorporating `aligner` in the near future.

Referring to **Figure 2.2**, the white diamond arrow connecting `cross-isorank` and `SBMLConnector` to `chequersGUI` means that `chequersGUI` is **aggregated** by the two classes. This is similar to a composition, except `chequersGUI`

is not required for `cross-isorank` and `SBMLConnector` (via the `main` class, which includes some basic code that tests the `SBMLConnector` functions) to work properly; this denotes the ability for the `carmat` and `aligner` modules to be run separately from a command line. When the user starts the Chequers application, the `mainApp` class is called, which simply initializes and displays the `chequersGUI` class on screen as the GUI.

The GUI has been tested as working on GNU/Linux and Windows operating systems. The `carmat` and `aligner` modules work fully via command line on GNU/Linux, Windows and MacOSX.

### **2.3.6 License**

Chequers is released under the GNU General Public License (GPL) which makes it fully FOSS-compliant.

## Chapter 3

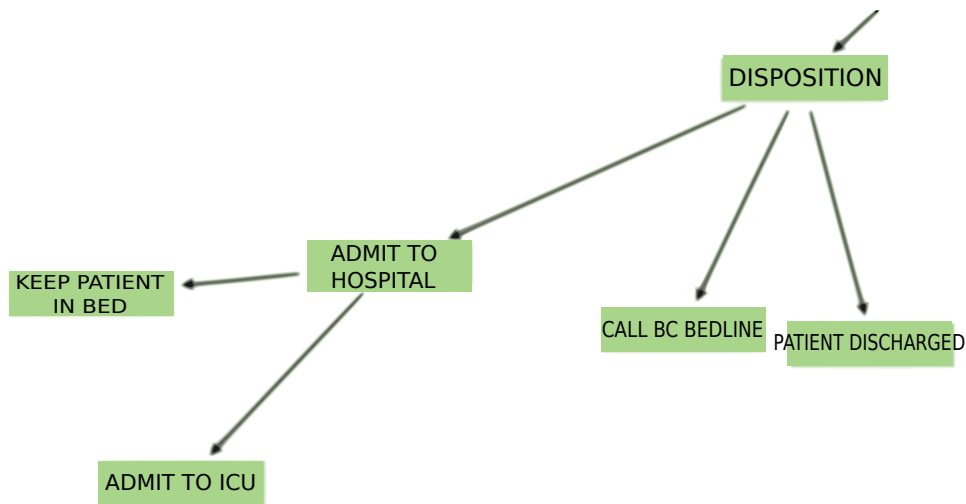
# Results, Discussion and Future Directions

### 3.1 Results

#### 3.1.1 Viability of a Curated SBML-based Care Map Document Standard

We were able to apply the SBML document standard as a viable option for digitizing care maps in a computable, sharable manner. This is supported by the SBML layout tool's ability to render an image of the SBML care map that is equivalent to the initial care map used for the translation. **Figure 1.2** shows what a generic care map looks like, while **Figure 3.1** shows a portion of the care map generated by Chequers. The difference here is that it was first manually translated into SBML and then rendered through the SBML layout tool. The full care map was too large to include here. The SBML and non-SBML maps both contained the same information. The major difference is that **Figure 3.1** was produced by computational methods, while **Figure 1.2** was drawn manually. This demonstrates that the SBML document format was viable for storing care map data electronically.





**Figure 3.1:** A portion of the same care map, curated with controlled vocabulary and rendered with the SBML layout tool directly into SVG format.

### 3.1.2 Alignment Results

The goal here was to try and use Chequers' `aligner` functionality with a sample input of two curated care maps, Hospital 1 and Hospital 2. The interaction and similarity input data were generated and analyzed through the `aligner` module, which contains the pairwise IsoRank implementation known as `cross-isorank`.

Our `aligner` was designed to accept care map input and produce similar output to the proprietary implementation, so it produces two raw data output files (formatted in a particular way for ease of reading), `match-scores.txt` and `cluster-output.txt`. The latter file shows the output when `cross-isorank` aligns two care maps. Both files were produced without error when we tested alignment of same two care maps: Hospitals 1 and 2. **Figure 3.2** shows that `aligner` was able to obtain cluster data for the Hospital 1 vs. Hospital 2 pairwise alignment. The clustering results shown in **Figure 3.2** are taken directly from the `cluster-output.txt` file, but the layout has been changed and colour added to clearly mark the boundaries between the clusters. The file also provides information on the input constants used to produce the clustering.

The IsoRank algorithm considers two pieces of information when calculating its match scores and clusters: the actual data in the nodes (i.e. in our case, the vocabulary and phrases used to describe care steps) and the shape of the pathways (in our case, how the care map nodes are connected to each other). In doing so, it requires three important constants: alpha ( $\alpha$ ), beta 1 ( $\beta_1$ ) and beta 2 ( $\beta_2$ ). These three constants are user-manipulable and affect the algorithm's output (Singh et al., 2008). The  $\beta_1$  constant is utilized in IsoRank's multiple alignment which was not implemented, since we only built a pairwise alignment module. Thus, we only need to consider the other two constants. The  $\alpha$  constant is a number between 0 and 1 which tells IsoRank how to weigh the two pieces of information it uses to compute the match scores: information within the nodes and the shape of the care map. The match scores, which are calculated by comparing each node from one care map to each node in the second care map, are then used in computing the clusters. An alpha of 1.0 means IsoRank will only look at the shape of the map, while an alpha of 0.0 means that IsoRank will only consider node content (Singh et al., 2008). By setting it to 0.6 in **Figure 3.2** we are asking IsoRank to weigh the two qualities relatively equally, with a slight edge toward the shape of the map. The 0.6 number was chosen through trial and error and it demonstrated the optimal output, as discussed further in **Section 3.2.2**. The  $\beta_2$  constant is used when the algorithm decides if a certain node is suitable for inclusion into the cluster it is currently trying to grow (Singh et al., 2008). For the alignment in **Figure 3.2**,  $\beta_2$  was set to 0.8, which roughly means that any new node that IsoRank is examining should have at least 80% of the value of the maximum match score already present in the cluster.

Constants  
Alpha: 0.600000  
Beta 2: 0.800000  
-----

If you see 0 next to the vocabulary, that vocab came from: ./hosp1.vocab  
If you see 1 next to the vocabulary, that vocab came from: ./hosp2.vocab



**Figure 3.2:** Results of aligning Hospital 1's and 2's care maps as produced by aligner. In total, using the input constants that we did ( $\alpha = 0.6$ ,  $\beta_2 = 0.8$ ), five clusters were generated.

What **Figure 3.2** shows is that IsoRank is able to make some sense of our care map data, which is not its normal use case. We discuss the actual implications of the alignment in **Section 3.2**, that is, what the clustering actually means for our use case.

## 3.2 Discussion and Implications

This project tries to bridge the gap between Bioinformatics and electronic health (e-Health) and defined the new field of “Translational Informatics.” Electronic decision support systems are one of the active research areas within e-Health, and will be implemented in care facilities as the natural progression from manual decision making. While the final decision is still up to the human care givers, decision support systems can aid in the process by presenting viable decisions to the care givers in an accessible manner. Decision support systems can also aid in standardizing care across multiple care facilities by illuminating strengths and weaknesses in per-institution care practices (Van den Berg and Visinski, 1992). It is the latter fact that is the underlying motivation for this project. CPG analysis is a form of decision support because it forms a positive feedback loop between implementation and guideline structure. Since CPGs are borne of evidence and implemented in the hopes of improving care, the implementation can be studied in order to elucidate strengths and shortcomings in the CPGs, forming said positive feedback loop. Thus, the decision making happens at two levels: first at the implementation level whenever a patient-centric care decision must be made (visualized as branch points in care maps), and secondly at the policy level, when the aforementioned positive feedback loop can drive both CPG research and implementation simultaneously.

Instead of developing a brand new electronic decision support system, we settled on the modular software design approach to tackle just the care map analysis problem in the hopes that these modules could be incorporated into future decision support engines that contain functionality for other areas of e-Health, such as storage and retrieval of Electronic Health Records (EHRs) or EMRs (Gaddis et al., 2007). In this manner, CPGs or care map analysis and decision-making can potentially be tailored towards individual patient parameters in the future.

The novel application of Bioinformatics in both care map standard development as well as analysis shows that its paradigms are applicable in clinical medical management. Bioinformatics has always concerned itself with crunching and calculation of massive data sets of nucleic and amino acid sequences, sometimes spanning terabytes across large clusters of computers. Care maps are miniscule in comparison (an SBML file describing a care map, on average, takes about fifty kilobytes of space on computer storage media), and so the need for processing power to perform the alignments should not be a concern. However, obtaining sensible output through the biological pathway analysis was critical, as it would verify that our translational approach worked. The IsoRank alignment algorithm was designed with both structure (biological pathways) and content (amino acid sequences) of the actual input data in mind (Singh et al., 2008), but we were able to translate vocabulary terms in care maps into protein analogues and still produce an alignment that made sense, using Singh et al.'s IsoRank algorithm. This shows that IsoRank can accommodate different types of input data, as long as one adheres to the input file formats required by the algorithm in order to compute its alignments. The algorithm just needs to be “tricked” into accepting different data that resembles its normal input.

### 3.2.1 Notes on XML and Sharability

By creating a document standard through SBML, we assume we are gaining sharability above and beyond what is provided through the other document formats normally used to express CPGs, but this is a networking and collaboration issue that is far more complex and requires much further research and software development. What exactly is sharability, and how is sharability of a care map in SBML different from, say, sharability of a JPEG image? At first glance the two cases seem to be identical since both file types can be physically shared between computers, printed out as hard copies and can be understood by care givers. Both are viable for transmission over the Internet using common protocols such as Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP) or most commonly, electronic mail. Thus, they are protocol-agnostic and *are* valid media for physically sharing CPG

data. However, by defining a document standard using XML-based technologies such as SBML, we can derive a much deeper implementation and meaning of the concept of sharability.

### 3.2.1.1 The SBML Layout and Render Extensions

XML is the parent language for a wide variety of document standards and schemas, and this project will benefit from this fact in the future. By using SBML here, we used a text document, in essence, to describe a series of processes and relationships between nodes in the care map. XML allows us to define our own schemas and structures, and SBML, being a derivative, conforms to XML specifications (Hucka et al., 2003). XML schemas are also conducive to extensions, which are peer-reviewed extensions of the initial standard that provide added functionality. This furthers the notion of sharability in a document standard, as people can add functionality to the standard without affecting the original document. For example, SBML has a layout extension for visualizing biological pathways. The extension stores information about graphical symbols, symbol placement, arrows, text placement, fonts and all other important visualization-related parameters (Deckard et al., 2006). The extension was incorporated into Chequers' `carmat` module; in addition to storing reaction information, `carmat` writes some basic layout information to the SBML document. Of course, since the SBML document is generated on-the-fly as a care giver adds vocabulary and reactions to it, it is impossible to predict exact placement of nodes and arrows ahead of time. This is where the external layout generation tool came in. The layout tool uses the information stored in the layout portion of the SBML document in order to convert the somewhat-cryptic SBML document into crystal clear images with arrows and nodes instantly recognizable. Furthermore, it figures out the layout on its own, that is, the exact coordinates of each graphical element. The layout extension and render functionality are built into `libsbml` (Bornstein et al., 2008) and it was a simple matter to add the extra lines of code such that Chequers knew how to produce the basic layout information required by the layout tool. The only issue with this is that the layout changes each time the tool is run, often requiring multiple renderings before

obtaining a suitable layout. Thus, the care map document is already more than basic SBML; it now includes the layout extension information, and this modification was achieved with relative ease. This also means that, in the future, if any existing changes need to be made to the schema, it is much easier to retrofit existing documents than recreate them from scratch using the newer version of the standard. Such facility is unavailable with static documents like JPEG.

### **3.2.1.2 Real-Time Collaboration and Revision Control**

While the classical sharing model of transmitting care maps across the Internet via email or some other protocol is valid, as noted above, we can leverage other XML standards in this regard. Take the case of collaborative care map development at an ED. In the classical model, one person might develop the initial draft on a single computer and then send it for review via email. A second person would make changes or additions and email it back to the first person, or to the rest of the care map development team. As this process repeats, we begin to see a dangerously long thread of emails and one can quickly lose track of which version of the care map is the latest. In an alternate example, perhaps the single care map document (a Microsoft Visio document, for example) is stored on a network drive as part of a Local Area Network (LAN). In this case, the file is modified by the care map team and the modified version is saved using a different filename, prefix or suffix to denote the new version; the more dangerous case would be when the file is overwritten with the changes. In these cases, the important thing is that the team is unable to work on the file concurrently. One person must finish with the changes and then send the new version around before someone else can do the same. If more than one person works on changes, it is impossible to merge these together using this sharing model.

To promote true real-time collaboration, there must be some form of revision control (Boxwala et al., 2001). Revision control systems, also known as Software Configuration Management (SCM) systems, allow concurrent editing of documents and maintain an exact history of every change made to the document. Thus, one

can revert back to a previous version in the event of a catastrophic error and multiple team members can edit documents together while still working remotely. Revision control systems work best with text documents, and not with “binary” files such as JPEG, Microsoft Word or Microsoft Visio documents, since it is easiest to compare text rather than pixel data, fonts, layout and other parameters. One such popular SCM is Subversion which has all the features described (Pilato et al., 2008). XML formats shine here as they are text-based. Thus, SBML care maps are highly conducive to revision control and satisfy Boxwala et al.’s requirement. An SCM module that leverages a popular FOSS SCM such as Subversion can be incorporated into Chequers in the future. As a side note, Subversion was used in this project to maintain all the source code as well as the  $\text{\LaTeX}$  documents used in writing and typesetting this thesis (C++ and  $\text{\LaTeX}$  source files are text-based documents).

### 3.2.2 IsoRank’s Viability in this Use Case

IsoRank was designed to perform pairwise and multiple alignments of biological pathways (Singh et al., 2008). In order to evaluate whether IsoRank could take our care map data and produce alignments, we had to first ensure it could handle a pairwise case before throwing multiple care maps at it. Armed with this “test pairwise first” methodology and faced with the sudden urgent situation of having to write an IsoRank implementation ourselves, it became necessary to write the code for the pairwise alignment while making provisions for the multiple alignments such that we could extend the code easily at a later date.

Looking at **Figure 3.2**, the mere fact that five clusters were generated immediately shows that the two care maps do have functional similarity, which can be expected since they are both sepsis care maps. Setting  $\alpha$  to below 0.5 (that is, emphasizing node vocabulary similarity) quickly reduced the output cluster numbers to one or two. This implies that, to understand the functional similarities between two care maps, it is important to also consider care map shape. Here we define a greater number of clusters as meaning a better alignment between two care maps. This seems arbitrary until we realize that, in order to gain useful quantitative insight



into care map alignments, we need a comfortable number of clusters that contain a few elements, rather than one cluster with entire care maps or numerous clusters with one element each. Only then can we ascribe any sort of functional similarity to sets of care map nodes. Similarly, 0.8 was a threshold value for  $\beta_2$  that produced a reasonable number of clusters with a manageable number of nodes in each cluster. So it seems as though `aligner` finds a cluster number of five to be the ideal value for the alignment between the two hospitals.

Referring to **Figure 3.2**, Cluster 2 shows some expected results. We see that “IV antibiotics given” and “See Triage Nurse (TN)” for both care maps show up in this same cluster. This means that, while the vocabulary was the same, the shape of the care maps around these nodes was also similar, that is, there was a functional similarity in the placement of these steps. Care map steps that come before and after these nodes are similar, so we can deduce that Hospital 1 and 2 both adhere to the sepsis CPGs in similar fashion at this point in their care implementations. Yet the phrase “MD orders tests, IV fluids, diagnostics” in Hospital 2’s care map was clustered in Cluster 3, and the same phrase from Hospital 1’s care map was placed in Cluster 4. At first glance, this seems strange because the vocabulary is the same. However, by setting the  $\alpha$  at 0.6, we had told `aligner` to give a little more priority towards the shape of the maps; seeing this separate clustering, we can infer that the care map shapes are different around this node in the two maps. This means that the care steps taken before and after these nodes are potentially very different in each ED. Cluster 5 seems to be a strange potpourri of seemingly unrelated vocabulary. “Patient discharged” is a step that is encountered at the very end of a care map, while “Patient waits for Triage Nurse” is usually a step that occurs early on in sepsis care. The only way to interpret these two nodes being clustered together is that the shape of the care maps around these nodes is in fact similar. We must consider the incoming and outgoing connections from these nodes. Thus, Cluster 5 is probably an outlier and can accidentally lead to false interpretation. This also implies that IsoRank’s results are more functionally significant *away* from the ends of the care maps and closer to the *centre*.

Thus, IsoRank seems to be a valid computational approach for conducting alignments on care maps. This discovery is a pinnacle of this project, as it opens the

door to standardized, quantitative analysis of patient management procedures and CPGs, supplementing the positive feedback loop established by evidence-based medicine.

### **3.2.2.1 Big Hammers for Small Nails**

It is worth noting here that biological pathway alignment algorithms are designed to work on huge sets of data, since pathways can often contain hundreds of thousands of nodes and reaction steps. Our care maps do not contain more than thirty nodes so the use of a pathway alignment approach could be perceived as akin to using a large hammer to affix a small nail. It is recognized that other approaches exist that may be more suited to tackling care map alignment. The results and discussion regarding the use of IsoRank in this project are based on an experientially based selection stemming from the discovery that a pathway markup language (SBML) could be directly applied to the description and storage of care maps. This led to the automatic thought process to attempt a pathway alignment approach, thus the exploration of IsoRank as a viable algorithm. Also, if in the future, as clinical care maps become more sophisticated and increase in the number of nodes, this pathway alignment approach would not present a ceiling effect as to how many nodes can be accommodated. This is because, even if care maps approach nodes in the thousands, biological pathway alignment algorithms are habitual in dealing with millions of nodes.

While the experience in this thesis sheds important light on this issue of care map comparisons, the modularity of the software development allows future flexibility: should a more suitable alignment algorithm be found, it can silently be swapped with IsoRank without cascading changes to the user level.

### **3.2.3 Musings on Granularity**

Since care maps are generated by humans, the human interaction context is critical to understanding the quantitative data that is derived from care map analysis.

Manual translation of care maps exposes the issue of granularity, that is, how much information does each node in a care map capture, and how does this differ between EDs? For instance, one ED may record a care step as “Physician orders antibiotics and tests,” whereas the same node might be broken into two discrete steps: “Physician orders antibiotics” and “Physician orders tests” at a second ED. In this case, breaking a care step into two alters the structure of the care map, which will confuse an algorithm like IsoRank, since it is unaware that the two steps in the second care map correspond to the single step in the first. This will cause a difference in clustering and matching of nodes between the two care maps. Defining and controlling the level of detail in each node is difficult since care maps are created by humans. Furthermore, in the latter care map, the two care steps may be reversed, that is “Physician orders tests” may come before “Physician orders antibiotics” or vice versa. For an algorithm like IsoRank that looks at the shape of the care map as well as node vocabulary, such a change can alter clustering and match scoring. Thus, the ordering of nodes can also play a part in affecting the final cluster results and this should be taken into account in future evolution and development of this software and care map comparison approaches.

## **Chapter 4**

# **Future Directions**

### **4.1 Comparison of Multiple Care Maps**

Our project to date was limited to studying the performance of a single pairwise alignment between the two hospitals' ED care maps. IsoRank is capable of multiple alignments, and so we must conduct more comprehensive tests in the future by aligning more than two care maps simultaneously and seeing if we can obtain sensible data. Our test data set was limited to two care maps, but many more will be available over the coming months as more EDs submit their sepsis care maps. Only then we will be sure that this analysis approach is scalable. This also means that we need to build up our controlled vocabulary to accommodate novel situations and care practices of other EDs.

### **4.2 Scoring Matrices**

Since CPGs are designed to be flexible around individual patient parameters, they are a step towards providing highly personalized medicine. With personalized medicine comes personalized decision-making. Since decisions are expressed as

branch points on CPG care maps and can have varying importance, it may be necessary to develop a more complex **scoring matrix** for the alignment algorithm. A scoring matrix can be generated by analyzing patient parameters and will allow us to flag certain portions of care maps as being of higher priority, or higher score (Needleman and Wunsch, 1970). The algorithm will take this into account during alignment, providing consensus data that may be more in tune with individual patients. While IsoRank does not operate in this way, any future alignment algorithm that utilizes prioritization or scoring matrices in its alignment generation might be able to provide such personalized results. Again, the modularity of Chequers allows for easy replacement of the alignment algorithm when the inevitable advances in pathway alignment algorithms are made, and so it allows for painless updating and modernization of the software as required. In our project, it would probably be wisest to develop said scoring matrix with the help of the care givers who utilize the CPGs in practice, as they will have first hand information on care step priorities.

### **4.3 Improving the Document Standard's Sharability**

Having defined a new standard, we are in danger of tying the standard too closely with the Chequers software; for if this were to happen, the SBML care map standard would be no better than Microsoft Visio's document standard. We would be forcing care map developers to use the Chequers software to create, visualize and share care maps in the new SBML standard, much like it is imperative that users have access to Microsoft Visio to view the Visio documents. True sharability should ensure that our document standard maintains most, if not all of its functionality on a wide variety of platforms; it must be platform-agnostic in order to gain acceptance from the community of CPG developers. This means that, much like a JPEG image, our SBML document should ideally provide a graphical view without requiring the external layout tool, that is, there should be a graphical aspect built into the standard. Imagine being able to double-click on the SBML document and seeing an image instead of XML text, just as one would do with a JPEG. Such an extension may be possible in the future with the use of the SVG image format. SVG is a vector graphics format, meaning there is no loss of quality during resiz-

ing and everything in the image can be described mathematically by vectors. SVG also has the added benefit of being a derivative of XML (Eisenberg, 2002)! Since most modern operating systems already know how to interpret an SVG image (that is, double clicking the image opens it in the appropriate image viewer software), perhaps the SVG rendering information can be combined into the remainder of the SBML document. The image viewer would perhaps parse only the SVG-relevant portions, yet the same document can be read as normal by Chequers (which only looks at the SBML portion). More research into this possible combination of standards must be done, as this will allow the user to simply double-click this new SBML-SVG chimæric document and view it without the need for Chequers. SVG is also supported by most modern Web browsers, so this new chimaeric document can easily be published on a website, should the need arise. Like every other XML document, the SVG is a text file, which means it takes up a tenth of the storage space normally required by binary image formats such as JPEG. As another side note, the graphics in this thesis were all rendered as SVG images to provide maximum print quality.

We are getting closer to realizing the sharability potential of our care map document standard, however, there is one extra component that can be added to this melting pot to deal with the issue of emailing files back and forth. While SCMs take care of this to some extent, it is more a system of controlling development and is not easily accessible to the general public; this makes sense as the public may perhaps be allowed to view the care maps without having access to editing them. There needs to be a simple method of broadcasting the care maps without requiring end users to install SCMs or even Chequers. While the chimaeric SBML-SVG document can be easily accessed via a web browser and this may serve as a useful way of publishing care maps, it often excludes mobile users. Clinicians and other ED care givers are quickly becoming adept at using the newest smartphones, netbooks and tablet computers, and these machines cannot be ignored as platforms for sharability. However, it is not always feasible to use a Web browser to retrieve care maps. Instead, it would make more sense if the care giver were “paged” on their mobile devices whenever a new care map became available; or if the care givers need to browse through a repository of care maps, they should be able to do so

without web browser access. XML potentially holds a key here as well.

A “Web feed”, or “syndication”, is the latest way to broadcast updates to a website or a web service and has become the de-facto standard for sharing information available on web logs (blogs). Two protocols, Really Simple Syndication (RSS) and Atom Syndication Format (Atom) were developed for the purposes of feeding content to users. On the user’s end, there exists a “feed reader” or “aggregator” that knows about a particular blog (the user provides the feed Uniform Resource Locator (URL) to the aggregator). Each time the blog author posts a new article, the feed readers at the user’s end pick up the new content, much like email. Without having to visit the website itself, the user is often presented with a “teaser” of the content and can choose to visit the main website in a web browser to view the whole article (Hammersley, 2005). Perhaps, then, RSS or Atom syndication can be utilized along with the care map standard. Conveniently, both RSS and Atom are XML-based standards. Perhaps a triple chimaeric document standard that sports SBML, SVG and RSS elements can be created to maximize sharability. Chequers should, of course, contain the necessary syndication modules in the future as well. If Chequers is available to the user then all functionality should be available through this single interface.

### **4.3.1 Thoughts on Mobile Chequers**

Modern smartphones and Internet tablets running BlackBerry OS, Android, Maemo or iPhone OS are now the norm, and it is conceivable that medically-relevant applications such as Chequers can be deployed to these platforms. As their processing power and features increase, it is likely that a full mobile complement to Chequers can be created for these devices without much hassle. The GUI would have to be rewritten to fit the smaller screen, but the inner workings and algorithms can remain intact.

Smartphones and tablets also have the added advantage of “push” alert features, where notifications (for new email, for example) are sent directly to the device. This is in contrast with the syndication approach discussed in the previous section,

where a feed reader needs to poll the feed at intervals; while mobile devices do have aggregator applications, they also have a “push” feature, whereby the devices can be paged as soon as new care maps become available. The user does not have to use a feed reader to poll the feed each time.

Mobile devices, therefore, present an attractive platform for Chequers because they enhance sharability by making the application’s suite of features available to clinicians on-the-go.

We have seen that XML holds the key to improving sharability, and the final care map document standard should include these elements in order to be attractive to care givers for adoption. Utilizing the aforementioned XML standards means that end users do not have to go out of their way to incorporate this new standard into their CPG and care map development practice. By using ubiquitous technology, we ensure that cross-platform portability and accessibility are maximized; a standard is no good if it cannot be easily accessed by everyone.

#### **4.4 Vocabulary Control and General Aesthetics**

The typical biological pathway contains nodes, just like care map nodes, except each node is usually an amino acid sequence describing the protein that is involved at that step in the pathway. When IsoRank examines pathways, it looks at node similarity, that is, it uses BLAST scores to determine how closely proteins from one pathway are related to proteins of another pathway. It makes sense to perform a BLAST alignment for amino acid sequences, but in our use case, the node data consists of sentences and phrases in the English language. English sentences cannot be aligned using any sort of sequence alignment, as they contain semantic data where the meaning of the sentence is greater than the words used in the sentence. Furthermore, multiple sentences can mean similar or same things. Each care facility that creates a care map could end up with identical care steps, only described using different vocabulary.

Vocabulary control was a problem for us, because we needed to translate sentence similarity into something akin to BLAST scores in order for IsoRank to work. We



chose to use a controlled vocabulary, whereby we essentially request each care facility to stick with a core set of vocabulary to describe common steps. Some of the vocabulary we chose is elaborated in **Figure 2.1**. When we set out to translate Hospital 1's and 2's care maps, we tried to stick with the controlled vocabulary terms as much as possible. Only when there was something unique to one of the maps did we add new vocabulary. This ensured that IsoRank had some node similarity data with which to work.

However, in the future, there needs to be a more flexible alternative. This is where ontologies can aid us greatly. OWL or some other semantic framework can do the bulk of the deciphering of semantic information within sentences. Thus, when we receive two care maps that use different words to mean the same thing, it is just a matter of using the ontology to flag the two sentences as identical, rather than restricting vocabulary on our end.

This leads into a discussion on the user interface. Chequers' current testing GUI is a minimalistic reflection of its features. It does not bore well to maintain this user interface during deployment to ED teams, as some of the details of the inner functionality of SBML or `aligner` have not been hidden from the user. A good GUI should encapsulate all functionality as succinctly as possible, hide the inner details of its implementation and still have some visual bells and whistles for enhancing the user experience.

## 4.5 Future Clinical Trials with Chequers

The pinnacle of the future direction for this project would see Chequers being implemented in EDs so that they can begin generating care maps through the software system and utilizing the new document standard. Clinical trials would generate CPGs implementation data for two important studies: the breadth study across different EDs, and the longitudinal study over time within a single ED. Along with the other future implementations and enhancements discussed in this chapter, Chequers will be able to provide a unified platform for creating, modifying, storing

and evaluating care maps in EDs and establish a new field of Translational Informatics that utilizes Bioinformatics and Systems Biology in conducting e-Health, e-Medicine and Health Informatics research.

## Chapter 5

# Conclusion

In order to properly evaluate CPG effectiveness across multiple implementations, we found it necessary to develop a computational approach. In order to perform computerized analysis, we needed to ascribe quantitative measurements to the CPG implementations. By utilizing Bioinformatics concepts, we developed a document standard for care maps, which were the most computationally viable CPG implementations available to us. This allowed us to digitize care maps and pass them through our own cross-platform implementation of the IsoRank pathway alignment algorithm.

By developing a care map document standard and analysis methodology, we have expanded the future potential of the project. We aspire to have this platform supporting evidence based CPG development and implementation, thereby facilitating physicians and teams to learn from each other. This team-based, collaborative learning could be supported by the alignments and evolution of care maps between their home and other EDs. Furthermore, this learning would be quantitatively driven, which means exact numerical values can be ascribed to comparisons rather than simple qualitative measurement. We hope Chequers and the associated technologies and techniques that were researched in this project can help EDs improve team-based care over time, thereby contributing to the improvement of our

health system for optimal clinical care based on the best prospective evidence in a timely manner.

# Bibliography

- (2006). *A Classification for Comparing Standardized XML Data*. → pages 29
- Advani, A., Lo, K., and Shahar, Y. (1998). Intention-based critiquing of guideline-oriented medical care. *Proc AMIA Symp*, pages 483–487. → pages 1, 3, 20
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410. → pages 28, 39
- Baca, M. and Swetland, A. G. (1998). *Introduction to Metadata: Pathways to Digital Information*. Getty Publications. → pages 7
- Bates, D. W., Ebell, M., Gotlieb, E., Zapp, J., and Mullins, H. C. (2003). A proposal for electronic medical records in u.s. primary care. *Journal of the American Medical Informatics Association*, 10(1):1–10. → pages 12
- Bell, D., Layton, A. J., and Gabbay, J. (1991). Use of a guideline based questionnaire to audit hospital care of acute asthma. *BMJ (Clinical research ed.)*, 302(6790):1440–1443. → pages 20
- Bernstam, E., Ash, N., Peleg, M., Tu, S., Boxwala, A. A., Mork, P., Shortliffe, E. H., and Greenes, R. A. (2000). Guideline classification to assist modeling, authoring, implementation and retrieval. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pages 66–70. → pages 20
- Bornstein, B. J., Keating, S. M., Jouraku, A., and Hucka, M. (2008). Libsbml: an api library for sbml. *Bioinformatics*, 24(6):880–881. → pages 22, 26, 34, 48

- Boxwala, A. A., Tu, S., Peleg, M., Zeng, Q., Ogunyemi, O., Greenes, R. A., Shortliffe, E. H., and Patel, V. L. (2001). Toward a representation format for sharable clinical guidelines. *Journal of biomedical informatics*, 34(3):157–169. → pages 1, 3, 12, 13, 16, 17, 19, 20, 26, 29, 30, 49, 50
- Cary, M. P., Bader, G. D., and Sander, C. (2005). Pathway information for systems biology. *FEBS Lett*, 579(8):1815–1820. → pages 29
- Cass, S. (2002). Free as in freedom: Richard stallman’s crusade for free software [book review]. *IEEE Spectrum*, 39(6):56–57. → pages 31
- Chen, M. and Hofestaedt, R. (2005). An algorithm for linear metabolic pathway alignment. *In silico biology*, 5(2):111–128. → pages 21
- Cioffi, J. and Markham, R. (1997). Clinical decision-making by midwives: managing case complexity. *Journal of advanced nursing*, 25(2):265–272. → pages 8
- Cole, L., Lasker-Hertz, S., Grady, G., Clark, M., and Houston, S. (1996). Structured care methodologies: tools for standardization and outcomes measurement. *Nursing case management : managing the process of patient care*, 1(4):160–172. → pages 1, 3, 15
- de Lusignan, S., Stephens, P. N., Adal, N., and Majeed, A. (2002). Does feedback improve the quality of computerized medical records in primary care? *Journal of the American Medical Informatics Association*, 9::395–401. → pages 12, 15
- Deckard, A., Bergmann, F. T., and Sauro, H. M. (2006). Supporting the sbml layout extension. *Bioinformatics*, 22(23):2966–2967. → pages 48
- Eisenberg, J. D. (2002). *SVG Essentials*. O’Reilly & Associates, Inc., Sebastopol, CA, USA. → pages 56
- Gaddis, G., Woods, T., and Patel, S. (2008). 117: Efficacy impacts of an evidence-based algorithmic approach to the treatment, matched to presumed underlying cause of nausea and vomiting, in the emergency department. *Annals of Emergency Medicine*, 52(4):S78. → pages 9, 15

- Gaddis, G. M., Greenwald, P., and Huckson, S. (2007). Toward improved implementation of evidence-based clinical algorithms: clinical practice guidelines, clinical decision rules, and clinical pathways. *Academic emergency medicine : official journal of the Society for Academic Emergency Medicine*, 14(11):1015–1022. → pages 8, 9, 13, 16, 46
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. M. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 edition. → pages 34
- Gauges, R., Rost, U., Sahle, S., and Wegner, K. (2006). A model diagram layout extension for sbml. *Bioinformatics*, 22(15):1879–1885. → pages 22, 34
- Gillespie, C. S., Wilkinson, D. J., Proctor, C. J., Shanley, D. P., Boys, R. J., and Kirkwood, T. B. L. (2006). Tools for the sbml community. *Bioinformatics*, 22(5):628–629. → pages 22
- Golden, T. M. and Ratliff, C. (1997). Development and implementation of a clinical pathway for radical cystectomy and urinary system reconstruction. *Journal of wound, ostomy, and continence nursing*, 24(2):72–78. → pages 3, 4
- Hammersley, B. (2005). *Developing feeds with rss and atom*. O’Reilly. → pages 57
- Ho, K., Marsden, J., Jarvis-Sellinger, S., Sweet, D., et al. (2009). A collaborative quality improvement model and electronic community of practice to support sepsis management in emergency departments: Investigating care harmonization for provincial knowledge translation. Successful Announcement April 2009. → pages 23
- Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., , the rest of the SBML Forum:, Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J. H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le Novere, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R.,

- Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., and Wang, J. (2003). The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531. → pages 35, 48
- James, P. A., Cowan, T. M., Graham, R. P., Majeroni, B. A., Fox, C. H., and Jaén, C. R. (1997). Using a clinical practice guideline to measure physician practice: translating a guideline for the management of heart failure. *The Journal of the American Board of Family Practice / American Board of Family Practice*, 10(3):206–212. → pages 15, 20
- Lambrix, P., Tan, H., Jakoniene, V., and Strömbäck, L. (2007). Biological ontologies. pages 85–99. → pages 29, 30
- Lovejoy, L., Bussey, C., and Sherer, A. P. (1997). The path to a clinical pathway: collaborative care for the patient with an ostomy. *Journal of wound, ostomy, and continence nursing*, 24(4):200–218. → pages 3, 4
- McDonald, C. J. (1997). The barriers to electronic medical record systems and how to overcome them. *Journal of the American Medical Informatics Association*, 4:213–221. → pages 12
- Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453. → pages 27, 55
- O’Neill, E. (2000). Utility of structured care approaches in education and clinical practice. *Nursing Outlook*, 48(3):132–135. → pages 1, 3, 4, 5, 7, 8, 15
- O’Neill, E. S. (1994). The influence of experience on community health nurses’ use of the similarity heuristic in diagnostic reasoning. *Scholarly inquiry for nursing practice*, 8(3). → pages 8
- Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Commun. ACM*, 15(12):1053–1058. → pages 32



- Picard, K. M., O'Donoghue, S. C., Young-Kershaw, D. A., and Russell, K. J. (2006). Development and implementation of a multidisciplinary sepsis protocol. *Crit Care Nurse*, 26(3):43–54. → pages 1, 3, 9, 10, 12, 13, 15, 17, 69
- Pilato, C. M., Collins-Sussman, B., and Fitzpatrick, B. W. (2008). *Version Control with Subversion*. O'Reilly Media, 2 edition. → pages 27, 50
- Pinter, R. Y., Rokhlenko, O., Yeger-Lotem, E., and Ziv-Ukelson, M. (2005). Alignment of metabolic pathways. *Bioinformatics*, 21(16):3401–3408. → pages 21
- Singh, R., Xu, J., and Berger, B. (2008). Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences of the United States of America*, 105(35):12763–12768. → pages 22, 27, 28, 30, 38, 39, 44, 47, 50
- Smart, J., Hock, K., and Csomor, S. (2005). *Cross-Platform GUI Programming with wxWidgets*. Prentice Hall. → pages 32
- Smith, T. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197. → pages 27
- Strömbäck, L. (2006). A method for comparison of standardized information within systems biology. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 1603–1610. Winter Simulation Conference. → pages 25, 29
- Strömbäck, L. and Lambrix, P. (2005). Representations of molecular pathways: an evaluation of sbml, psi mi and biopax. *Bioinformatics*, 21(24):4401–4407. → pages 22, 25, 26
- Van den Berg, R. and Visinski, P. (1992). Decision trees in icu. *The Canadian nurse*, 88(1):28–29. → pages 15, 46
- Venner, G. H. and Seelbinder, J. S. (1996). Team management of congestive heart failure across the continuum. *The Journal of cardiovascular nursing*, 10(2):71–84. → pages 15

Wächter, T., Wobst, A., Schroeder, M., Tan, H., and Lambrix, P. (2006). A corpus-driven approach for design, evolution and alignment of ontologies. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 1595–1602. Winter Simulation Conference. → pages 29

Wang, S. (2003). A cost-benefit analysis of electronic medical records in primary care. *The American Journal of Medicine*, 114(5):397–403. → pages 12

## Appendix A

# A More Formal Definition of Care Maps

### A.1 A Basic Care Map

Let us take sepsis as a model, where the patient requires care from a Triage Nurse (TN), a Bedside Nurse (BN) and a physician. Each care giver has a role-specific protocol:  $P(TN)$ ,  $P(BN)$  and  $P(Physician)$ . The care map,  $CM(Sepsis)$  can therefore be defined as:

$$CM(Sepsis) = P(TN) + P(BN) + P(Physician)$$

**Definition 2.** The care map is an amalgamation of caregiver-specific protocols.

Let us now examine a scenario where this care map would be utilized. When a potential sepsis patient walks into the ED, the relevant staff can refer to  $CM(Sepsis)$  (in its flowchart form) and follow the steps specific to their role and expertise in order to successfully triage and care for the patient until disposition. A common step in sepsis care is for the patient to receive antibiotics at some point during the care procedure (Picard et al., 2006). Now, imagine a second patient walking in with the

same symptoms, except this second patient has an allergy to one of the antibiotics in the bolus administered to the first patient. Here is where care maps and protocols, in the currently defined form, show their shortcomings. Neither a protocol nor a care map makes any assumptions about the patient, that is, all patients would be treated exactly the same regardless of their symptoms or other parameters. For the second patient, following the exact steps as outlined by  $CM(Sepsis)$  would be catastrophic. If there were a way to tailor care maps to individual patient symptoms, care givers could provide standardized care that would be flexible enough to handle a wide variety of patient cases.

### A.1.1 Incorporation of an Algorithm

Let us see how our above scenario changes when  $Alg(Sepsis)$  is applied in practice in lieu of  $CM(Sepsis)$  (**Definition 2**). When the first sepsis patient walks in, the care givers follow  $CM_{upstream}(Sepsis)$  and reach the branch point described in **Definition 3**. When this patient's parameters are identified as  $PP_1$ ,  $CM_{PP1}(Sepsis)$  is applied to this patient. Now, if the second patient (with the antibiotic allergy) walks in, the care givers notice that the antibiotic allergy is part of the parameters, and therefore, these patient's parameters,  $PP_2$ , are different from  $PP_1$ . The care givers can swiftly switch to  $CM_{PP2}(Sepsis)$  downstream of the branch point (as per **Definition 3**) and provide this second patient with the same level of care as the first patient. Care map nomenclature ( $CM()$ ) can be used for each component of the algorithm because the steps downstream of the branch point can be thought of as either whole care maps or subsets of care maps pertaining to care procedures that must be implemented based on the decision made at the branch point.

$$Alg(Sepsis) = CM_{upstream}(Sepsis) + \begin{cases} CM_{PP1}(Sepsis) & \text{parameters} = PP_1 \\ CM_{PP2}(Sepsis) & \text{parameters} = PP_2 \end{cases}$$

**Definition 3.** An algorithm with a single upstream care map until a branch point. Now, when two sepsis patients walk in, the care givers follow a common upstream care map until the branch point where they must make a decision about which care map to use, based on the patients' parameters.