

Development of a Prostate Cancer Patient PSA History Calculator To Support Patient Follow-Up Scheduling Decisions

by

Tom Harding

B00120003

tharding@cs.dal.ca

Performed at

Radiation Oncology Department

Dickson Building, Capital District Health Authority

5820 University Avenue

Halifax, NS

In partial fulfillment of the requirements of the Master of Health Informatics Program,

Course HINF 7000 (Internship), Dalhousie University

Report of Internship for the period July 15 – December 31, 2009

Date Submitted: January 4, 2010

Acknowledgement and Endorsement

This report has been written by the author in partial fulfillment of the requirements of the Master of Health Informatics (MHI) Program at Dalhousie University. This report has not received any previous academic credit at this or any other institution.

I would especially like to thank Dr Rob Rutledge at Capital Health, who has been an absolute pleasure to work with on this project. I can only hope that my next work assignment will be with someone as cheerful and supportive. I would also like to thank Dr Grace Paterson for being so very helpful in my search for a work-term experience that was close to both the clinical world and the technical world. These are both arenas that interest me greatly and I enjoyed my time spent on this project because of it. I would also like to thank all my professors at Dalhousie University for sharing their knowledge that has been so interesting to learn and apply in what I am finding to be a remarkably worthwhile field of interest.

Tom Harding

Executive Summary

The primary goal of this Health Informatics (HI) work-term was to develop a software application which can be used by the Radiation Oncology Department of the QEII to predict the probability that a patient's prostate specific antigen (PSA) levels may rise by an estimated amount over a specified period of time. The data for these calculations was from a database maintained by this department containing about 2000 patient histories in MS Access. This application was created by the author under the guidance of Dr Rob Rutledge between mid July 2009 and the end of December 2009. All software development expertise, tools, planning, and host platform were supplied by the author to meet the requirements of this department. The work was composed of 6 phases:

- 1) **Topic Familiarization and Research**: Receive introductory clinical familiarizations from Dr Rutledge, and learn primary objective (a workable application) and secondary (long term plan) requirements. Search library sources for related publications. Meet department staff. Review format/content of patient charts. Meet Ron Dewar of Cancer Care Nova Scotia. Review epidemiology projects done using 'R'. Receive raw data files from Dr Wilke.
- 2) **Data Exploration and Refinement**: Prioritize data fields. Identify records belonging to Dr Wilke and correlate secondary file data to primary file. Review OPIS data from CCNS. Produce initial PSA behavior graphs. Experiment with regression analysis methods. Develop data clipping methods. Propose 3D nearest neighbor CBR approach. View data using WEKA.
- 3) **Algorithm Trials and Demonstrations**: Develop overlapping patient plots separated by risk levels, add start window filtering and end window filtering, then calculate percentage of PSA failures per grouping. Apply iterative processing of previous calculation to produce a probability graph over time. Rework to meet user expectations (month time scale, unique PSA failure plots, etc)
- 4) **User Application Integration and Testing**: Search for appropriate GUI application tools which can interface to 'R' routines, are supported by CDHA network PCs, can be applied quickly, and are zero cost. Develop PSA History Calculator GUI, integrate to 'R' routines and graphing output devices. Add patient charting feature and exact count displays. Test multiple use cases, debug errors encountered, compile delivery version.
- 5) **Product Documentation and Delivery**: Write a comprehensive user guide to explain tool use and dataset updating. Print and deliver hardcopy with all software products on CD.
- 6) **Planning Future Improvements**: Write technical proposal for delivery to BC hospital authority for request to use their 6000 patient dataset to augment our existing 300 patient set.

The resultant prototype created allows a user to select a patient risk level, their time of Radiation Therapy (RT), the present date, their present PSA level, a future follow-up appointment date, and an estimated future PSA level on that date. It will calculate the probability that they will exceed that future estimated level on that future date based upon approximately 300 eligible patient histories from the original dataset. It allows the user to see a composite plot of previous patients who were similar to the selected case and how many of those cases exceed the anticipated PSA growth. It also allows the user to see the probability that the entered case will exceed the future estimated level as a function of time between the present date and the date of the future follow-up appointment. As an additional feature it will specify which previous case histories did exceed the entered parameters to allow gathering of further details by comparing the chart of the present patient against the chart of those specific previous patients if desired. All computations and history plotting was performed using the statistical script language called 'R', and the program integration and graphical user interface (GUI) was constructed using a language called 'C++'.

Health care practitioners are not computer experts, nor should they ever need to be. Healthcare practitioners have some strong preconceptions about software and computer systems in general that will evolve gradually with time, patience, and experience. Health Informatics solution developers need to be equally open to change, new ideas, and unique ways of looking at problems, because only by combining our skills and knowledge effectively with one another can we achieve the best results. Solutions must "fit" the needs of the practitioners who know their subject matter best, and their patients as well. Nobody wants a "solution" that makes their goals harder to achieve.

A significant limitation of this tool is that it has only about 300 source cases to draw on to make decisions. The result is that it can give significant results in some of the most common case scenarios, but when a case scenario diverges from the typical path of progression, the results quickly become statistically insignificant because there are only a handful (or fewer) previous cases that have been similar. We have proposed to solve this by seeking significantly larger patient data history sets from other sources with similar treatment practices to those of CDHA. The first of these is a set of 6000 patient histories from a hospital in British Columbia. Acquisition of this data will allow for creation of a greatly enhanced source case dataset and logically should lead to improved result significance and accuracy.

Table of Contents

1. Introduction	7
2. Description of the Organization.....	10
3. Description of the Work You Performed at the Organization	11
Primary Work Role and Activities	11
Secondary Activities.....	12
4. Discussion on How Your Work Relates To Health Informatics	14
5. Discussion of a Problem Analyzed and the Corresponding Solution.....	16
Topic Familiarization	16
Data Exploration and Refinement.....	17
Algorithm Trials and Demonstrations.....	21
User Application Integration and Testing :	29
Product Documentation and Delivery	30
Planning Future Improvements	30
6. Conclusions	31
7. Recommendations	33
References :	34
APPENDIX A – Draft Proposal.....	36
APPENDIX B – Activity Summary.....	38
APPENDIX C – ‘R’ Script to create “Future Date View”	42
APPENDIX D – ‘R’ Script to create “Term View”	48
APPENDIX E – ‘C++’ “main” code for GUI Control.....	53

APPENDIX F – PSA History Calculator User Guide..... 57

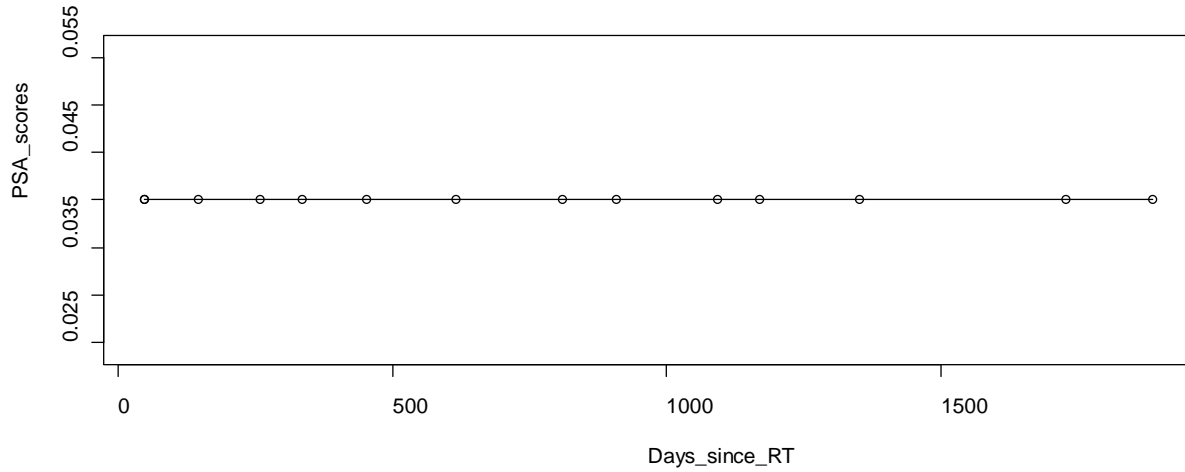
APPENDIX G – Further Data Request and Proposed Direction 74

1. Introduction

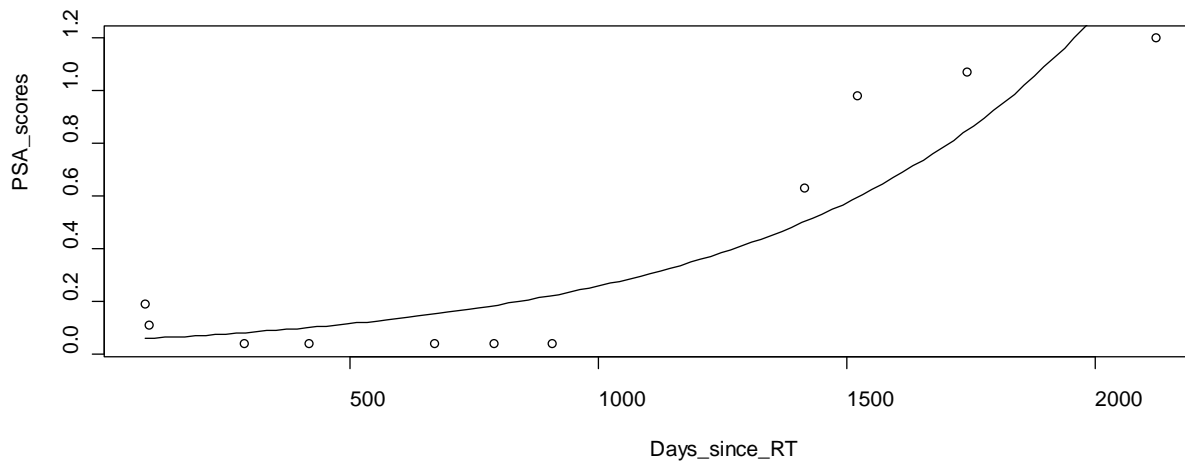
Prostate Cancer is diagnosed in about 18 thousand men in Canada every year. Unlike many other cancers however, prostate cancer grows quite slowly. The 10 year survival rates are in excess of 90% for this disease, and in those individuals who eventually have a recurrence after treatment, most will eventually die of other causes. The detection and treatment of prostate cancer requires careful detection, management and periodic follow up monitoring. For monitoring, prostate specific antigen (PSA) is measured as a cancer tumor marker in blood tests that often signals the onset or recurrence of this disease and it may be detected years before intervention is required. These long timeframes therefore create debate about the optimal follow up schedule for this population. At this time, more information is needed to develop one based on quantifiable objective evidence. A tool that can help predict the likelihood of recurrence would be a great help in determining the most reasonable frequency of PSA monitoring and in-office follow-up appointments that are needed.

Such a tool would be required to forecast when a patient's PSA level and growth rate are clinically significant in those patients with prostate cancer treated with radiation therapy (RT); with or without neo-adjuvant androgen deprivation therapy (NAAD). This would be particularly advantageous for patients who have a low probability of recurrence requiring intervention. Accurate PSA forecasting could reduce unnecessary follow up appointments and therefore reduce costs to our already over-taxed health care system.

Typical PSA levels in healthy males are nearly zero, but PSA blood levels can double and redouble repeatedly with the onset of cancer. The resultant PSA profile over time in an untreated case where cancer is present gradually becomes exponential. *(All PSA plots in this report were created by the author using the features of 'R' (cran.r-project.org 2009))*



A PSA Plot Indicative of No Cancer Regrowth



A PSA Plot Indicative of Cancer Regrowth

During this author’s work-term, we began with 10 years of recorded PSA test results for roughly 2000 patients as a basis for calculations. The goal was to generate a decision support tool to help practitioners determine what is an appropriate follow-up appointment schedule tailored for patients of various risk levels and treatment durations. The source data was collected by Dr Derek Wilke at the Radiation Oncology Department of CDHA Halifax. The requirements for the calculation results and data presentation needed by practitioners were chosen by Dr Rob Rutledge, also at the Radiation Oncology Department of CDHA Halifax. The data filtering and cleaning, investigations, software tools selection and acquisition, algorithm

development, implementation and testing were performed by the author between July 2009 and December 2009.

When work began, it was clearly reiterated that the primary desired outcome was to be a functional PSA history calculator that could be used at the Radiation Oncology Department of CDHA Halifax upon completion. This is also stated in the initial problem description, dated 21 June 2009, as included Appendix 'A'.

“Dr. Rutledge will approach the Informatics division of the Dalhousie Computer Science Department. The plan is to develop a computer program which will draw on the anonymous database to predict the probability of recurrence over a specified interval according to patient characteristics (eg. Initial prognostic factors, current PSA level, PSA doubling time if PSA rising, number of years in follow up etc).”

This author was well suited to this work because of his extensive experience developing and fielding software solutions, and this work was well suited for this author because it was planned on a protracted schedule which allowed other activities to be done concurrently both by the author and by CDHA. This work has been an illuminating practical exercise in numerous facets of Health Informatics (HI) including the collection of insight and experience from experienced practitioners, the need to understand clinical treatment practices and requirements, the application of data mining principles to 10 years of good-but-imperfect patient data, the choice and use of appropriate data and software tools on a minimal budget, the application of basic statistical principles on real world health data, and finally the algorithm development and testing itself. This report will detail the activities performed by the author during these activities which are primarily focused on fulfilling the primary goal mentioned above, and it will also include description of other activities which were performed by the author as a result of related and unrelated learning opportunities at this facility during this timeframe.

2. Description of the Organization

This work-term was conducted under the guidance of Dr Rob Rutledge, at the Radiation Oncology Department, Dickson Building 1st and 3rd Floor, Queen Elizabeth Health Sciences Center, Capital District Health Authority in Halifax, NS. This location handles over 16 thousand cancer patient visits per year. The department is responsible for numerous forms of patient cancer care, including radiation therapy for prostate, lung, head/neck, brain, gastrointestinal and breast cancer. Inpatient care by this department is conducted on 8A of the Centennial building at the VG site.

This organization functions as a portion of the greater Cancer Care Nova Scotia organization which includes both the mainland Nova Scotia (Capital Health Cancer Care Program) and the Cape Breton (Cape Breton Cancer Centre) services. The Capital Health Cancer Care Program portion extends well outside of the Halifax Regional Municipality and includes facilities in the Annapolis Valley, New Glasgow, and Yarmouth. The Cape Breton Cancer Centre is based primarily in Sydney, Nova Scotia but it also includes facilities in Antigonish and Inverness.

3. Description of the Work You Performed at the Organization

Primary Work Role and Activities

Health Information Analyst and Software Developer

My primary responsibility while working at CDHA was to fulfill the primary project requirement which was to “develop a computer program which will draw on the anonymous database to predict the probability of recurrence over a specified interval according to patient characteristics” (Rutledge/Wilke 2009)

During the course of this work it was necessary to complete several stages of topic research, data exploration, algorithm development and trials, user application integration and testing, product documentation, and future planning. The activity of integrating data from the CCNS Database could not be performed because the data that we received from them was completely dissected on an attribute by attribute level and there was no way to correlate which belonged to each patient record. The activity of trialing a more sophisticated data manipulation was tentatively formulated based upon a Case Based Reasoning (CBR) approach, however for reasons that will be discussed later in this report that approach was set aside for now.

This included performing literature searches through Dalhousie libraries and elsewhere for supporting project information sources, and later sharing that information with other department students (clinical). The author gathered client (user) requirements and expectations, summarized them to get concurrence, and proposed technical solution approaches. It was also necessary for the author to choose, locate and supply the necessary software tools within budget (\$0) to perform the project work (this department has no existing software development experience or tools).

As a working prototype took shape, the author iteratively demonstrated its function, gathered feedback and made changes to accommodate needs/expectations. As is frequently the norm in situations where the client is not familiar with the software development process, it was necessary to explain the many steps that exist between development of a first “somewhat functional” version and a final fully functional/tested version. Only the most trivial of software solutions work completely and flawlessly with

the first revision, as any experienced software developer can attest. A more detailed account of the project design and development are presented in section 5 of this report. A brief overview of the day by day work activities performed by the author is included in Appendix 'B'.

Secondary Activities

During the term of this work, opportunities arose on occasion to take a short break from focus on the PSA History Calculator development and to experience other things. An account of those follows:

Grand Rounds (Thursdays at Noon)

From time to time, CDHA has "Grand Rounds" in the ballroom of the Bethune building. At these events notable researchers and specialists present their work for review and questions by the audience attending. Although the detail was often "over my head" it was nonetheless fascinating to see departures from previously accepted standards of care and the results achieved during trials by these presenters on subjects such as Thyroid Cancer (for example).

Detailed Tour of New Charles V Keating Emergency Center

The author had an opportunity in October to spend a couple hours doing a very thorough tour of all the health information systems in the new Charles V Keating Emergency Center at the Halifax Infirmary site. This new center is a vast improvement over the previous one at this site in both size/volume and in systems installed. One of the most notable was the new Workstations on Wheels (WoWs) that now bring CDHA computer information systems right to the patient bedside for display and entry of all patient information.

Appointment Wait Time Guarantee Project

The CDHA Radiation Oncology Department is also in the midst of an overhaul of the way it organizes and schedules patient appointments in order to guarantee as minimal wait time for services as possible under present policy. This is a large project which includes many people both inside and outside the department and the author had an opportunity to review some of their work and attend a detailed meeting on the state and progress of this project.

Prostate Cancer Clinical Pathway Normalization Project

Under the guidance of Dr Raza Abidi, the author initiated discussions with this department to examine the possibility of developing a common clinical pathway mapping solution which would allow the smooth transition of patients at one location or jurisdiction to transfer to another. Such a system would track the steps, results and requirements completed for each patient and remap their progress based upon the new clinical pathway when they are moved. This would permit an open and transparent health care planning approach to patient treatment which might otherwise cause a patient to lose valuable time or even restart treatment when they are moved to another healthcare jurisdiction.

Volunteer At a Public Immunization Clinic

With the heavy burden placed on the healthcare system in the fall of 2009 H1N1 immunization activities, the author volunteered to assist at a CDHA public immunization clinic in Sackville, Nova Scotia. In this role the author helped direct patients, distribute information, gather consent and authorization forms, assist RNs with supplies, and clean up afterwards. It was a great opportunity to see how such an activity is conducted and gave information to consider for how a Health Informatics initiative might improve upon a system driven exclusively by pen, paper, and stapler. Perhaps in the future this could be changed to a Health Card, EMR, encrypted wireless networks, and barcode scanner based system.

4. Discussion on How Your Work Relates To Health Informatics

Health Informatics is concerned with the conventions, methods and mechanisms of gathering, manipulating, and disseminating health related information and knowledge for the purpose of improving health outcomes. Health Informatics is not specific to computer systems as is often perceived, however it is often related to computer applications as these are the most common form of health data processing systems at this time. "Health Informatics is as much about computers as cardiology is about stethoscopes"[Enrico Coiera 2003]. There are numerous aspects to Health Informatics, and they can be enumerated many ways. One of our early texts called "Guide to Health Informatics" by Enrico Coiera [2003] lists these aspects as:

1. *"Understanding the fundamental nature of health information and communication systems, as well as the principles that shape them"*
2. *"Developing solutions which can improve upon existing information and communication systems"*
3. *"Developing methods and principles which allow such solutions to be designed"*
4. *"Evaluating the impact of these solutions on the way individuals or organizations work, or on the outcome of their work"*

The work that has been accomplished on this work-term qualifies as a Health Informatics related effort due to its qualities meeting the criteria of aspects 1, 2, and 3. At the beginning of this work-term the author had very little knowledge of prostate cancer or the issues related to its detection, screening, treatment, and monitoring. The initial stages of this work were dedicated to learning more about this facet of healthcare both from the healthcare organization itself, and from books/articles from topic researchers. Prior to this work-term the Radiation Oncology Department had some gathered patient information in a MS Access database, but they had no way to query it to produce probabilities of future patients to exhibit particular PSA growth rates after treatment. Such systems do not currently exist to perform these predictive calculations in this particular way and so we have developed a new way to organize and present the information for review and uptake by the health professionals who need it to support their planning.

The work for this project included leveraging skills & information from several HI courses:

HINF 6030 HI Statistics: This course was absolutely essential to this task. Without this course the author would not have understood the principles of regression analysis, the importance of statistical power in calculations, or the important difference between independent events and dependent events.

HINF 6101 HI Flow and Use & HINF 6102 Hi Flow and Standards: This course content was valuable for interpreting the information that the author reviewed related to the wait time guarantee initiative, and how the information was being gathered and how it would be used as a basis for revising the established departmental standards.

HINF 6120 Clinical Care Fundamentals: This course was also absolutely essential to this task because the author does not originate from a health care background. There is so much terminology which is assumed to be common knowledge that the author would not have understood well without this course. Also it gives a footing in how to find out more about clinical issues or topics when you need more information.

HINF 6230 Knowledge Management for HI: This course was definitely very applicable to this task. In the long term the principles of this course would yield a superior solution to this problem. The problems which prevented this from happening in a meaningful way in this initial prototype are that we did not have a large amount of case data to use as a basis for a CBR type system, the principles of CBR are not understood by practitioners and they prefer a more statistical approach, and the time available for this work was very short and the requirement of completing a functional prototype before completion was clearly stated.

OCCU 6504 Measuring Health Outcomes: This course was useful because it explains the dangers of blindly accepting the numbers generated by testing, and practical ways to interpret them dependent on the population of interest. It is primarily focused on ordinal data testing and outcomes, but the value in being circumspect of source data can safeguard against drawing rash conclusions in many situations.

PSYO 6101 Computers & Instruments in Psycho Experiments: This course was invaluable for this task. The author would highly recommend this course for Health Informatics students because it introduces the 'R' language, which is both very powerful for data manipulation and statistical testing. It is also free to use under GNU license. The author would not have been able to complete this prototype in the short amount of time available without this course.

HINF 6210 Data Mining for HI: This was also an essential course to this work, and it would have been preferable to have completed this course prior to doing this project. Nonetheless the knowledge and principles of this course especially related to the preparation and surveying of data is a great asset in performing tasks similar to the work on this project.

5. Discussion of a Problem Analyzed and the Corresponding Solution

This work was performed as a series of phases beginning in July 2009:

Topic Familiarization

Initial readings and discussions with Dr Rutledge covered many aspects of prostate cancer condition and treatment. These included:

- Transition of PSA generation change from normal prostate cells to cancer cells
- The roles of biopsies and other tests.
- The decision process requirements to support a “treat for cure” plan
- The relatively slow growth and high 10 year survival rate
- The role of hormone therapy in conjunction with RT
- Debate over appropriate follow up scheduling
- PSA doubling time on case by case basis
- Dangers of neurological compromise or systemic obstruction
- Gleason score, and (Tumor/Node/Metastatic) TMN scoring
- Pain management
- DRE scheduling and necessity conjecture
- Effects of co-morbidities on treatment and outcomes
- The roles of biopsies and other tests.

Typical treatment regimes are for:

- Low risk: Radiation Therapy (RT)
- Intermediate risk: Neoadjuvant Androgen Deprivation Therapy (NAAD) & RT
- High risk: NAAD, RT, and Long Term Androgen Deprivation Therapy (LADT)

Risk levels are generally defined as:

	PSA	Gleason	Tumor	criteria
High	≥ 20	≥ 8	T3	ANY OF
Moderate				NOT High or Low
Low	≤ 10	≤ 6	T2a or less	ALL OF

Data Exploration and Refinement

On the 16th of July 2009, the author was given two data files extracted from the MS Access database use at the Radiation Oncology Department of CDHA. The first file was called "Patient_Data_July152009" and the second file was called "PSA_values_July152009". The first file contained anonymous patient data for 1910 patients. The second file contained 15252 PSA test results for these patients. The fields included for each patient in the patient_data file included:

chart	STAR	Dx_date	Tumour_Site
HCN	MSI	risk_strata	T_Stage2003
N_Stage2003	M_Stage2003	Initial_treatment	Research_study
Erectile_Function	death_date	PSA_at_consult	gleason_score
first_gleason_pattern	size1	size_of_mass	treating_rad_onc
hormones_and_ERBT	total_AD_duration	date_start_AD	date_finish_AD
Total_number_injections	post_prostatectomy	surgery_date	pstage_2003
positive_margins	rising_PSA_post_surgery	PSA_prior_to_salvageXRT	palpable_postRP_recurrence
date_start_RT	dose	fractions	Pelvic_lymph_node_XRTdose
Pelvic_lymph_fractions	fractionation	number_of_months	total_month_of_hormones
Brachytherapy	B_dose	B_months	disease_status
date_last_follow_up	F_U_plan	date_asymptomatic_recur	date_symptomatic_recurrence
biochemical	date_bone_scan_positive	date_urinary_obstruction	date_cord_com
biochem_date	date_local_recurrence	date_bone_pain	date_other_recurrence
recurrence_Tx_date	recurrence_Tx	cancer_death	worse_GI
worse_GI_lasted	worse_GI_resolved	worse_GU	worse_GU_lasted
worse_GU_resolved	field_size_length	field_size_AP	field_size_PA
adequate_FU	critical_FU	year_patient_added	referring_urologist
PSA_nadir	obstructive_urinary_symp	irritative_urinary_symp	local
bone	other	DOB	oldtreatment
treatment	number_biopsy_core_taken	number_cores_positive	percent_biopsy_area
code	diagnosing_hospital	LHRH_agonist	date_RadOnc_consult
target_date	priority	intent	tumor_site
Area.technique	position	cast_device	Planning_imaging
bolus	energy	date_start_chemo	Family_Dr
Medical_Oncologist	Nomogram_chance_recur	Signed_Consent_form	Total_dose_prescription_isodose
Pelvic_isodose_prescription	Boost_isodose_prescription	date_finish_RT	date_Simulation_RT
date_start_cycle1_chemo	date_start_cycle2_chemo	Treatment_modified	Hospital_name
Hospital_address	Hospital_country_Postalcod	Hospital_phone	Hospital_fax
Eligibility	Patient_phone_numbers	Patient_email	

The fields included for each patient in the PSA_values file included:

chart	date	PSA	Free_PSA
Free_to_total_ratio	minimum_PSA		

Dr Rutledge decided that we would only examine the patients of Dr Wilke initially, and so it was rather simple to use MS Excel to filter out all patients except those belonging to Dr

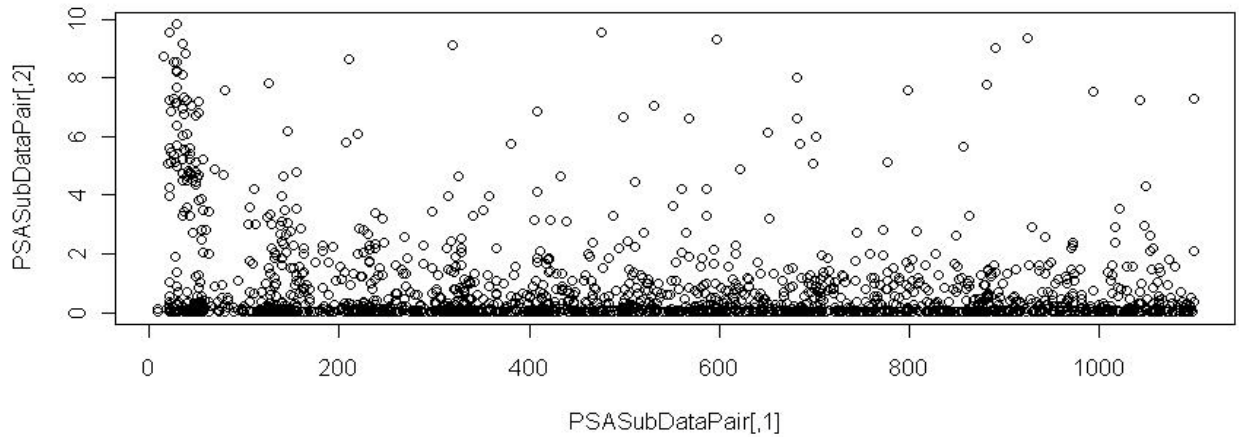
Wilke and then save this set of patients to a new file. The only difficulty was that there were multiple enumerated values in the attribute “treating_rad_onc” (treating radiation oncologist), which indicated Dr Wilke, so all were chosen as relevant. This newly created patient dataset belonging to Dr Wilke included 678 patients. The difficult portion of this stage was reducing the 15252 records in the PSA file down to include only those records which belonged to patients of Dr Wilke. This could theoretically be done using the filtering feature of MS Excel as was done for the patient file, but that would probably take about a week to isolate all PSA values for these 678, and this method would be highly susceptible to human error. Instead I wrote a program to parse the PSA data file and retain only those values which belonged to one of Dr Wilke’s 678 patients. This reduced the PSA file from 15252 records to 3192 records (about 5 PSA test results per patient).

The filtering program (written in ‘R’, a very powerful statistical language for manipulating tabular data):

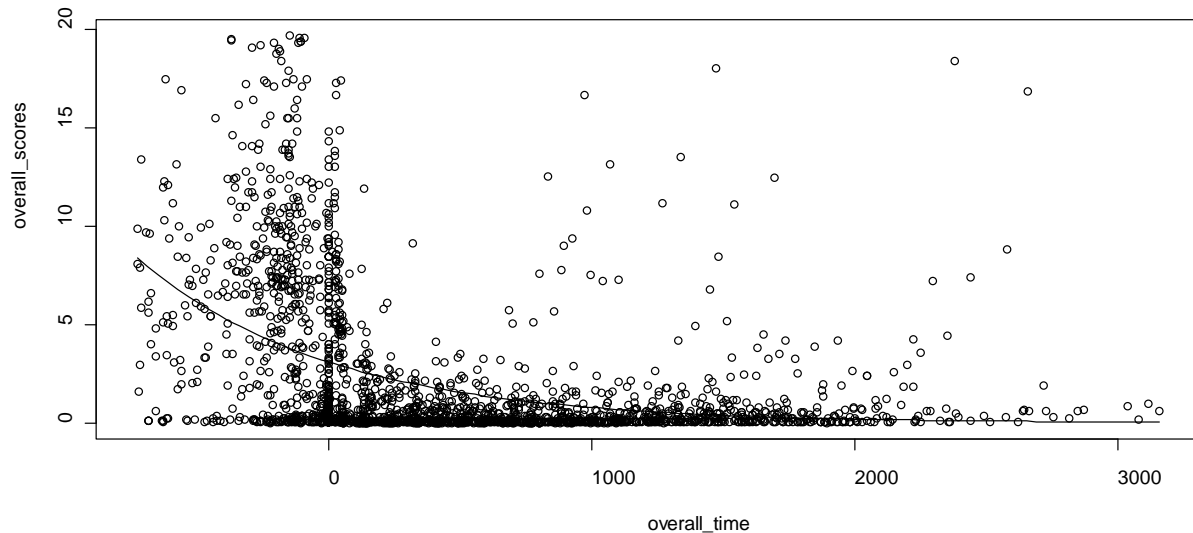
```
PatRawData <- read.csv(file="Patient_Data_DrWilke.csv",head=TRUE,sep=",")
PSARawData <- read.csv(file="PSA_values_July152009.csv",head=TRUE,sep=",")
PSA_DrWilke_Pats <- PSARawData [PSARawData[,1] %in% (PatRawData[,1]),]
```

A problem that was also noticed immediately was that all dates in these files were recorded in Gregorian calendar days (Eg 21 April 2003). This is fine for human readability, but it is a problem for computer processing because it is not easy to calculate the number of days between two Gregorian dates precisely (Eg 21 April 2003 to 17 October 2004) and allowing for leap days. The next step therefore replaced all date values in the patient information files with Julian calendar dates. The formula for converting Gregorian dates to Julian dates is “ $(367 * \text{YEAR}) - \text{INT}(7 * (\text{YEAR} + \text{INT}((\text{MONTH} + 9)/12))/4) + \text{INT}(275 * \text{MONTH}/9) + \text{DAY} + 1721013$ ”. There are several variants of this formula depending on which base calendar day is desired, but since this project only uses the number of days between Julian dates, they would all give the same result for our purposes. The Julian date for 21 April 2003 is 2452750. The Julian date for 17 October 2004 is 2453295. The difference between the two is now simple to calculate as 545 days. So that all patient histories could be chronologically compared side-by-side, the author decided to use their date of radiation therapy as day 0, and each date of PSA testing as an offset from that date (PSA testing prior to treatment would have a negative timestamp).

With all of the data filtered, and re-dated it was then possible to depict our PSA scores graphically to get a visual sense of what type of data distribution we would be working with. The first visual depiction created for this data was:



This preliminary look was created using only the majority of PSA scores between 0 and 10, plotted based only on 2 criteria (time since treatment (X axis) and PSA level (Y axis)) for patients up to 1100 days after treatment. The initial PSA scores near day 0 are elevated because they were recorded before the patient blood levels had subsided from their pretreatment levels. What this first glance shows is that PSA values fall quickly after treatment and generally remain very low thereafter. To get a better impression of how markedly different the PSA levels were before and after treatment, the data was re-plotted to include PSA scores up to 730 days before treatment:

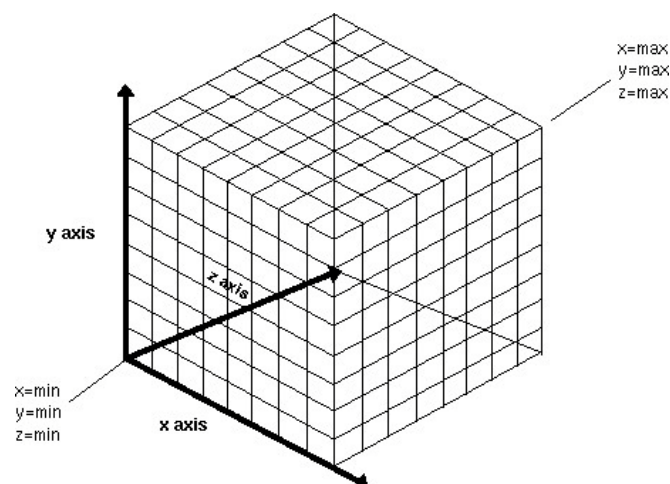


This second view more clearly shows the distinct difference in PSA scores before and after treatment (day 0). Using 'R' a least sum of squares regression line was superimposed over the data points to emphasis the distinct trend, which is not easily seen amid the very dense number of near zero PSA scores to the right of treatment date. If this were a set of disjoint data such as "number of cigarettes smoked per day before and after a stop-smoking course", we could use the regression line to provide an evidence based guess of how many cigarettes we might expect to see a member of this population smoked per day before and after the course. For estimating PSA however, there were a couple major obstacles which prevented simply plotting out the data and calculating a regression line. First, the regression line clearly shows that PSA tends to fall over time, not rise. This would infer we would never need to follow-up a patient because their PSA will continue to fall indefinitely. Secondly, these points are not independent of each other. This means that a pre and post treatment PSA curve for one patient can be distinctly different from another, and a rise in one score after treatment can indicate following score will also tend to rise. Using a population based multivariate linear regression (MLR) model was thus abandoned. It could be made to work with huge amounts of data, such that many patient subsets were created and each assigned an MLR model, but in this case if the data were divided in this way it would rapidly decrease the predictive power of any solution to be too low to produce reliable results.

Algorithm Trials and Demonstrations

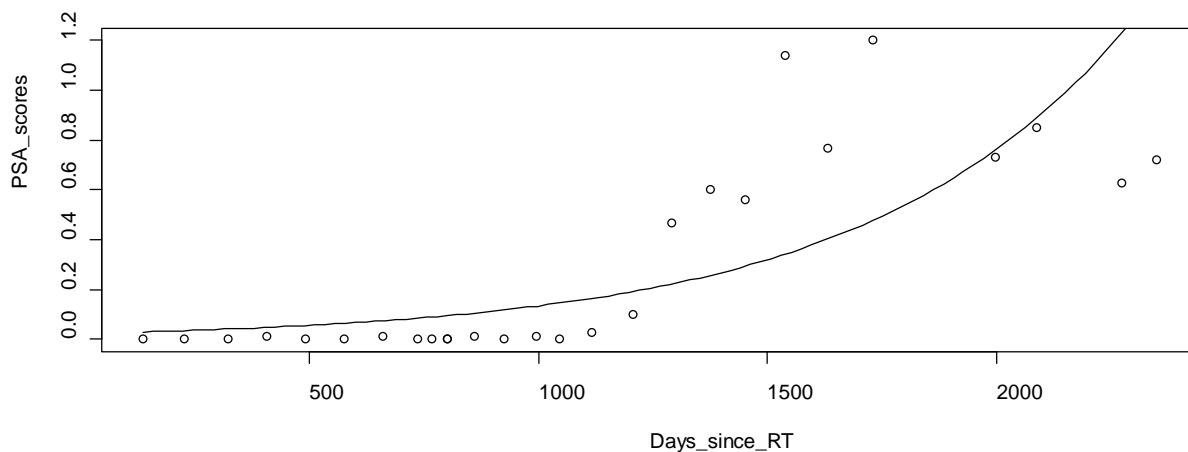
The early stages in any project include a need to establish the goals, appraise the assets available to attain those goals, and gauge potential solution approaches. As described above, we initially began with a basic explanation of prostate cancer, its characteristics, monitoring, treatment and follow-up. Next follows a thorough investigation of the dataset. In light of the primary objective of this task being to develop a tool to help practitioners choose an appropriate follow-up appointment schedule for patients based upon their condition and the objective case evidence of previous patients. One of the most promising avenues to fulfill this goal was to calculate the PSA growth rate (called PSA doubling rate) for patients as a function of their other characteristics using multivariate linear regression, and then use that growth rate in a time based formula to estimate their projected future PSA level, and to solve for the estimated amount of time that would pass between their present PSA and their chosen future PSA of interest. It was not known however, if the dataset could support this approach.

Another possible approach was inspired by the detailed explanation of PSA score behaviors in radiation therapy programs described in "*A Clinical guide to prostate specific antigen*" [Loughlin 2004]. In this book Loughlin gives numerous charts that give estimated PSA doubling times for patients of varying pretreatment attributes and treatment types. Using this approach it could be possible to model patient PSA growth by examining the post treatment PSA growth of many case histories, calculating their doubling time and then associating that doubling time with their other pretreatment attributes. With such a list of attributes and doubling times, a second algorithm could be used to model the doubling time attribute over the range of pretreatment attributes, or a near match doubling time could be selected from a case based store of calculated doubling times. A store of such values could be conceptualized as:

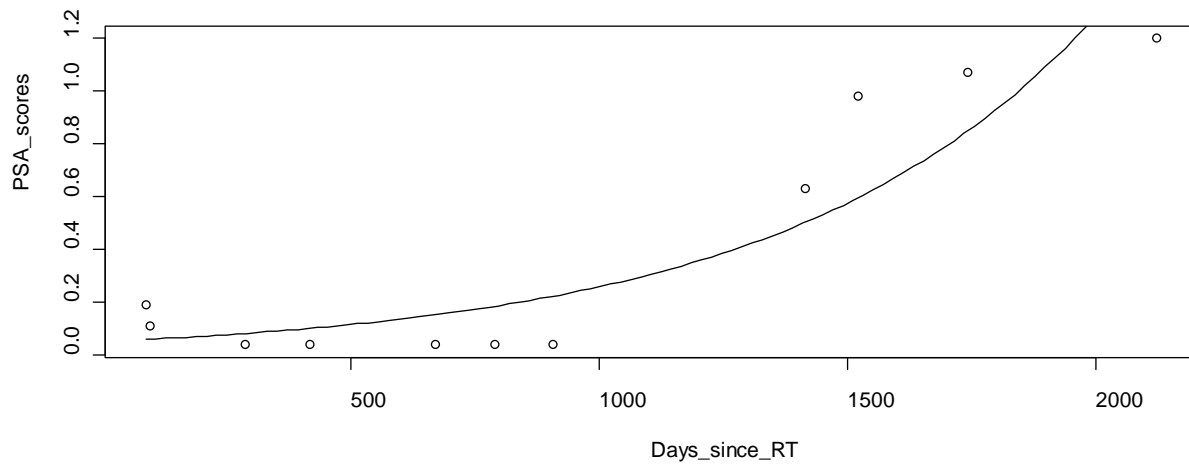


For instance we could generate a list of patient values for Pretreatment PSA, Gleason Score, Treatment Dose, and PSA Doubling Time. We could index our case results using X, Y, and Z for each attribute and then storing the PSA Doubling Time at that index. If we receive a new patient with the same attributes as an old patient, we could predict their PSA growth based upon the evidence from the previous matching patient, or if there is not an exact match we can choose a doubling rate for a previous patient that was a near match. If the source case history database is large enough we may have several previous matching previous cases to our new case and would have the luxury of picking the median PSA value from those results and a confidence interval for our projection. This approach would not have to be three input indices and one outcome index (PSA double rate in this case). It could have 1, 2, 3, 4 or more input indices. The tradeoff is that the more indices used, the more splits there are to find matching case histories and the less evidence there will generally be for each set of input indices.

The PSA doubling time from previous histories could be calculated by plotting their PSA behaviors and smoothing them with a regression line to remove excess sampling noise and then running a doubling time calculation on the results of the regression model. Some example plots for such cases and their regression lines would appear as such:



Case 1: Doubling Rate



Case 2: Doubling Rate

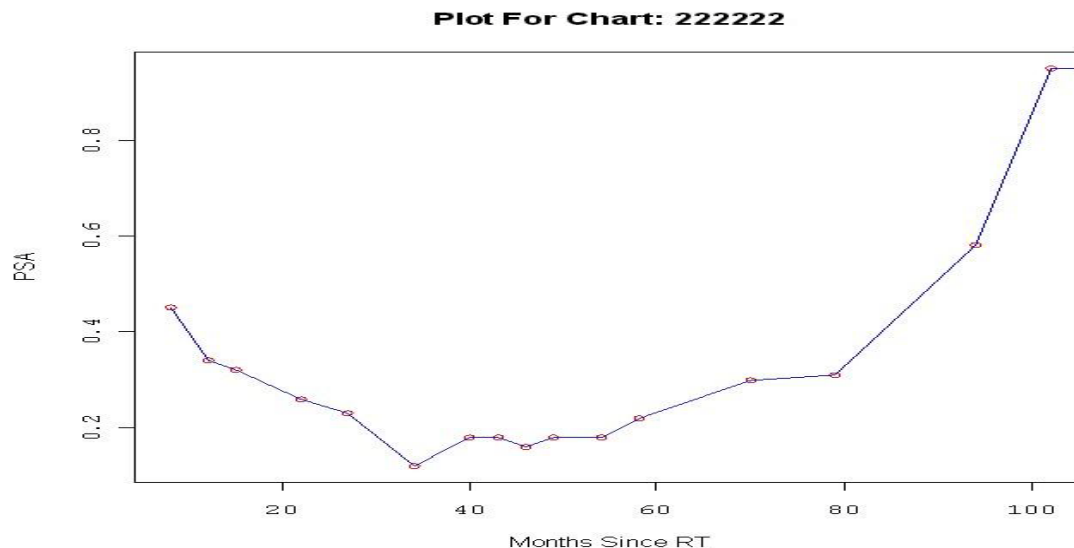
This seems a very interesting approach based soundly on Case Based Reasoning (CBR) principles. Its benefits are that it draws directly from previous patient histories, gives long term projections to allow early planning, and can be tailored to accommodate preferred indicators based upon the characteristics of the source data. Its drawbacks are that it requires a huge amount of source patient data and the algorithm for determining the nearest neighbor case in a 3 or higher dimension array become complex and time consuming quickly. This is an approach that the author would like to pursue at a future opportunity when sufficient source data is available. It was experimented with during this project work-term, but it was abandoned due to data, time and tools restrictions. We therefore turned to a more pragmatic approach.

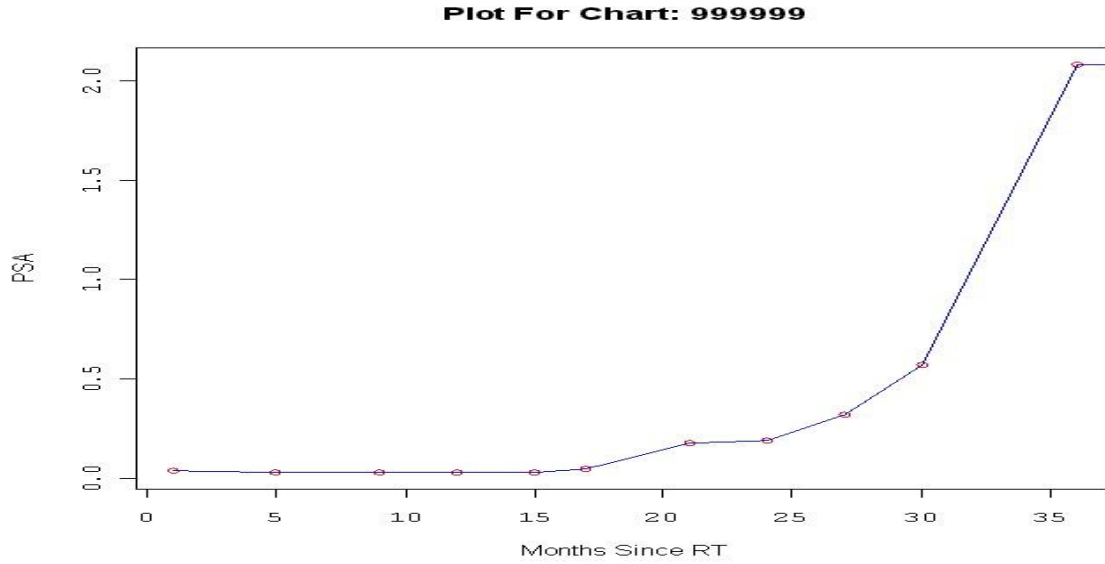
Implementing a Workable Solution In 'R' Alone

At this point it was decided we needed to proceed along a more pragmatic and expedient course to develop a solution with the time and resources available. To this end the patient indicators were trimmed to a minimum and processing was likewise scaled back to support a straightforward modeling solution. It was decided that it would be best to render a patient prediction based upon a minimum set of input data (thus subdividing the patient data a minimum number of times):

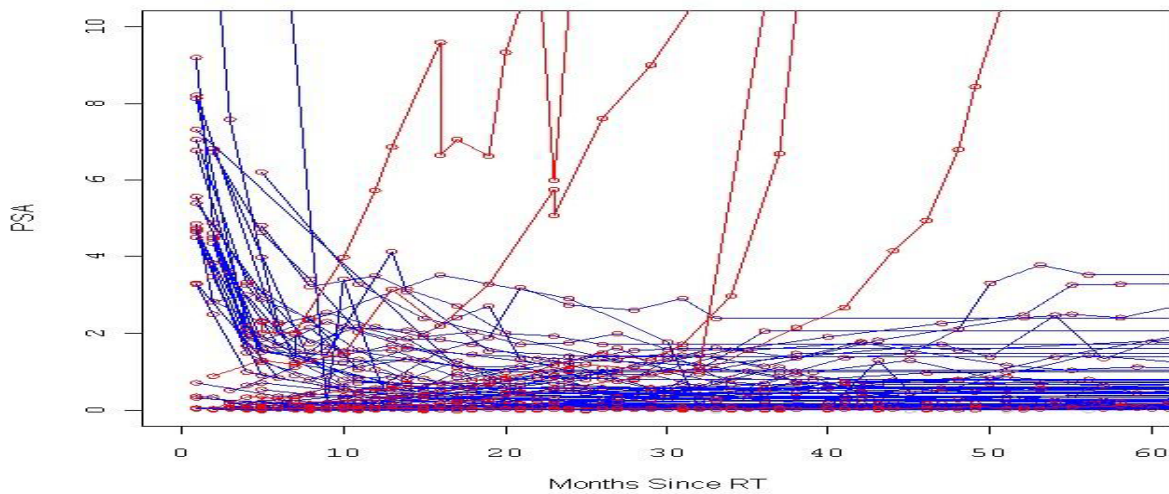
- Patient Risk Level
- Number Days Since Treatment
- Patient PSA Level
- Future Expected PSA Level
- Date of Future Expected PSA Level

With such a system it is possible to plot all existing patient histories, and at any point in time look for other patients who have in the past had similar PSA levels at the same time interval after treatment. Using those similarly matching previous patients it is then possible to find how many went on to exceed the future expected PSA level, and how many did not. This gives a ratio of how many can be expected to increase about the future expected PSA at a future expected date (ie, a future follow up appointment date). Straightforward plots of previous patient PSA histories can be individually plotted as in these two examples:

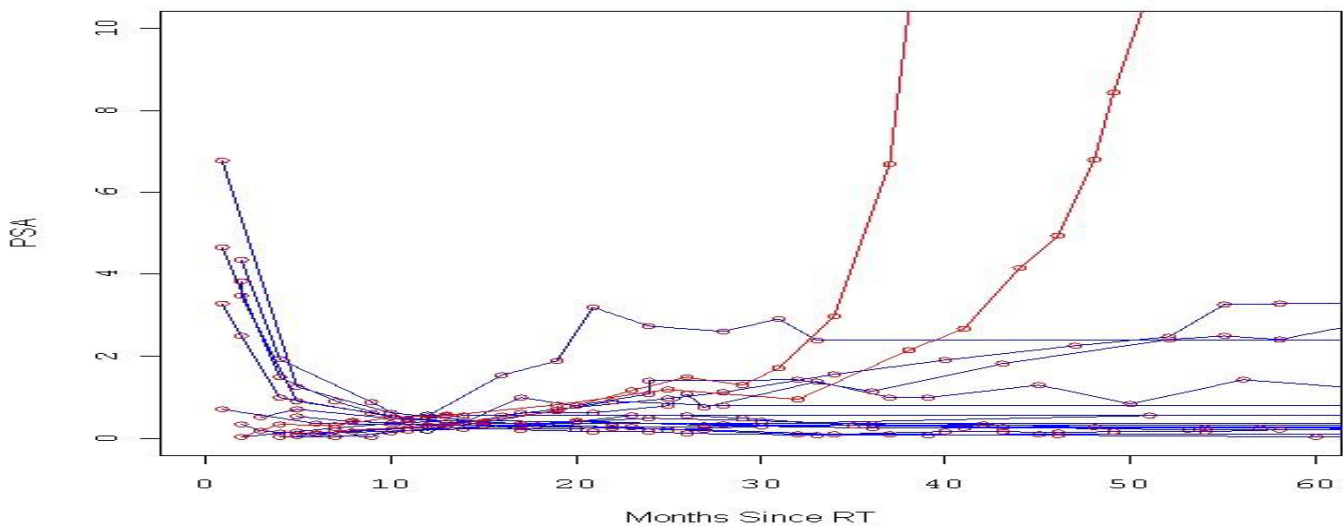




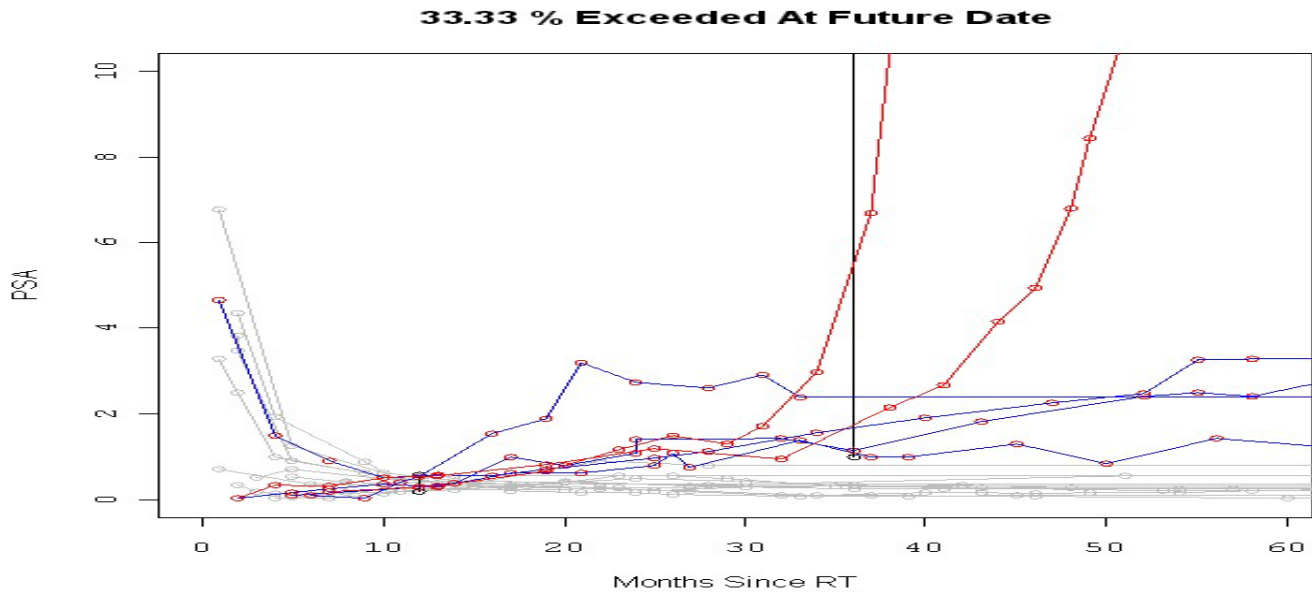
In order to create these plots, of course, each patient must have a significant number of PSA test results recorded. For this project’s projection calculations we chose to use all patients whose histories included at least 5 PSA history points following treatment. This reduced our patient history pool from 678 to 289 patients. There were many patients who had 4 or fewer points and it would be reasonable to assume some of these have had 1 or more new PSA test results since July 2009 and could be added to this modeling by drawing a fresh dump of patient history from the MS Access database. For the time being however we are still using the original data from July 2009. When we graph a large number of patients (all intermediate risk) simultaneously we get a large and unintelligible plot jumble that looks like this:



This again, does not tell us a great deal. It shows many patient plots starting at various PSA levels including many that are near zero post treatment and the majority remain very low over the span of plotted time (5 years in this case). For our purposes we do not want to see all patients at the same time, we only want to see those that are similar to our present case of interest. To do this we added a window of interest filter at the point in time matching the PSA test date of the present case of interest and we decided to retain history plots that were near the same PSA ($\pm 50\%$, minimum window of 0.2). This allowed us to look only at those cases that we are interested in for comparison to our present case. For a current case PSA of 0.4 at 12 months after treatment, this gives us a set of plots that looks like this:

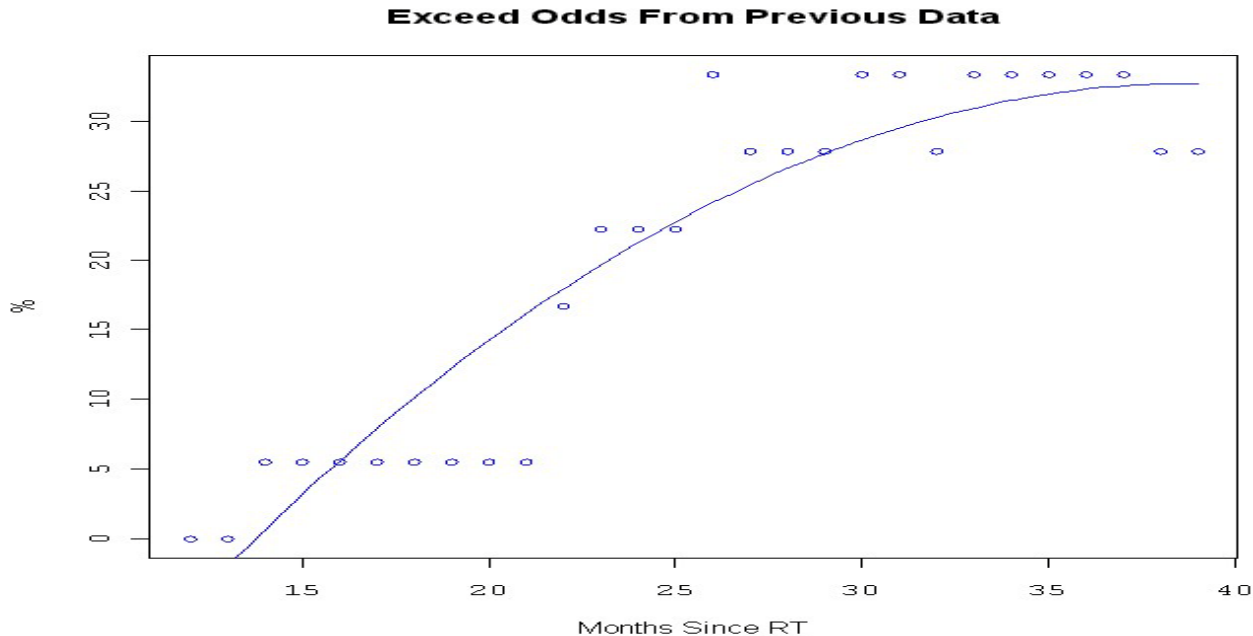


This step has given us a set of patient histories with comparable PSA to our present case of interest at the same point in time following treatment. It includes about 12-14 previous PSA histories. The question we may have at this point is how many of those previous histories that are similar to our present case eventually went on to exceed a PSA of 1.0 at 3 years after treatment? This answer is accomplished by adding a second window to the group of plots at the 3 year point that separates the histories that exceeded 1.0 from the ones that did not. In addition we can then count the number of cases that exceeded this threshold and divide that number by the total number of similar cases to find the percentage of cases that eventually exceeded our future threshold of interest. The resultant plot would then look like this (cases that did not exceed are plotted as grey background info only):



This view (called “Future Date View”) gives us some information that we can use to gauge our decisions for present day cases. It tells us that there were 18 cases that previously had a PSA of about 0.4 at 12 months after treatment (the start window of 0.4 +/- 50% is drawn in black on the plot). It also shows us that 6 of those cases went on to exceed 1.0 at 36 months after treatment (the end criteria window of 1.0 or more is also drawn in black on the plot). This gives a resultant PSA exceeded result of 33% in this case (6/18). The plots shown in grey did not exceed the threshold; the ones in color did exceed it. The plots in red rose as a result of PSA failure following treatment and as a result these cases were later changed to include long term androgen deprivation treatment (LADT info is stored in the PSA data file).

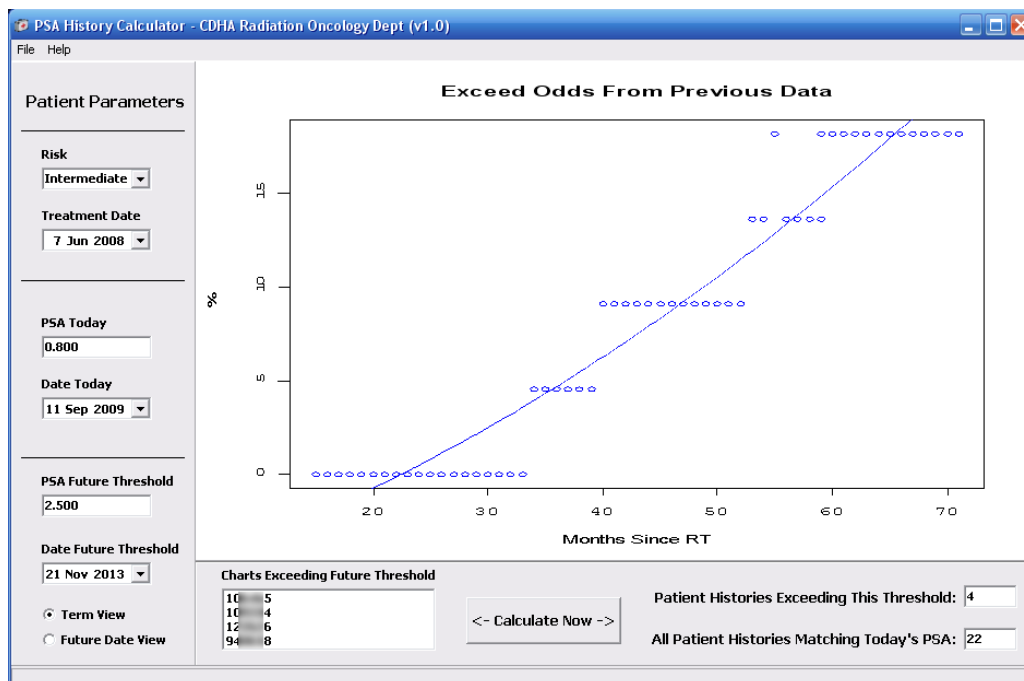
From a practitioner’s point of view this is interesting, but it only gives the PSA failure ratio at one point in time. The practitioner would prefer to see the PSA failure ratios at all points in time across a span of time during which they can choose how high a risk of PSA failure is acceptable. To do this it was necessary to calculate the PSA ratios that exceeded the threshold of interest for every month between the present point in time and the future point in time and then depict those graphically as well (called “Term View”)

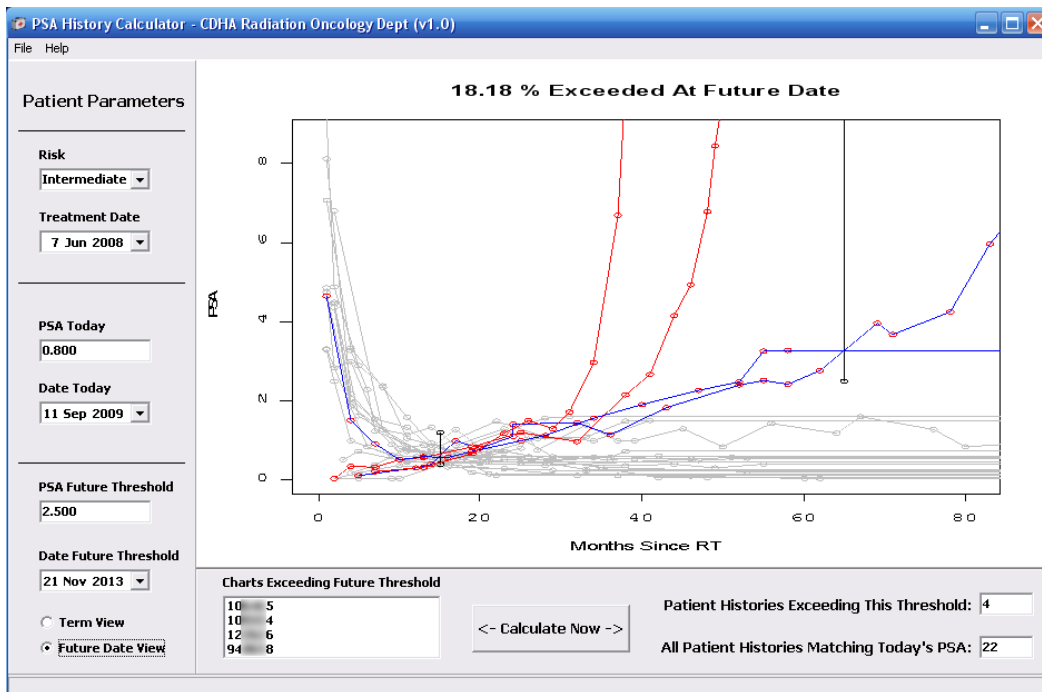


This is the final view of rates at which PSA levels have exceeded our threshold of interest (1.0) for all months from the present point (12 months) to the future point (36 months). A regression line is overlaid on the points to summarize the general trend. From this graph a practitioner could choose that they would only like a 10% chance that their present case may exceed a PSA of 1.0 prior to their next follow-up appointment. In such a case a scheduled follow-up appointment at about 18 months would be a good choice based on previous case evidence. The source code written to perform these functions is included in this report in Appendix 'C' and 'D'. At this point our solution, written in 'R' gives a solution that can process patient PSA history data and give a useful result, with one significant problem. 'R' was designed for manipulating data quickly, applying statistical methods, and outputting the results to a file on disk or in a display at the 'R' console. It was never intended for building graphical user interfaces (GUIs) or for ease of use by relatively inexperienced computer system users. It has very poor user interface ability and would definitely not be suitable for use by health practitioners who have more important things to worry about than "How do I get this script loaded into that console and where do I have to run it such that it will see my input files and create a result that I can see?".

User Application Integration and Testing :

With a functional solution in 'R', the next phase of the project became finding a way of presenting this ability in a way that made it simple and familiar to use. 'C++' was chosen as the language of development for this next phase for three primary reasons. First it is the gold standard of object oriented GUI development, it is versatile at interfacing to other languages and systems such as 'R', and thirdly it has an Integrated Development Environment (IDE) that is familiar to the author and thus allowed development of an enhanced solution in a short period of time. Specifically the IDE used was Borland Development Studio 2006 (C++). A simple GUI was created that allowed for input of present case parameters, patient risk strata, present case PSA level, specification of anticipated future PSA and date of occurrence, and the selection of the type of result depiction desired ("Future Data View" or "Term View"). Furthermore some additional display features were added that were not possible in the 'R' only version, this improved version gives exact counts of patient cases used for determining the results and thus implies the power of the calculation and it also gives the exact chart numbers of the patients which previously exceeded expected PSA growth levels such that a practitioner can look up previous cases for details of their specifics if they wish. The GUI layout and appearance is as shown (note chart numbers purposely have 3 digits blurred in these screenshots to hide their true value). To use this version the user simply needs to double click the icon added to their desktop, specify the case parameters, and press "Calculate Now". This GUI is a visual object created with Borland Developer Studio 2006, and the 'C++' code used to control this GUI is included in Appendix 'E'.





Product Documentation and Delivery

A more complete and detailed explanation of this PSA History Calculator operation and the requirements for managing its history dataset was compiled into a User Guide and delivered to CDHA along with the project software. For a detailed view please see the User Guide that was written and delivered to Dr Rutledge along with the software. It is included as Appendix 'F' of this report.

Planning Future Improvements

This project is an interesting beginning in a direction that has a long way to go to reach a mature conclusion. Data quality issues remain, as well as the potential for expanded case attribute matching, and confidence interval determination to name a few. This project has wrapped up for the moment at this point of looking for more supporting data and data processing methods and goals that may be achievable. This is discussed further in section 7, and the preliminary data request proposal created to address these issues is included in this report as Appendix 'G'. This request/proposal was written primarily by the author with some subject background provided by Dr Rutledge and a clinical student (Graham Cook).

6. Conclusions

During this work-term a functional PSA History Calculator was constructed to meet the needs for Dr Rutledge and Dr Wilke at CDHA. They have both expressed great satisfaction with it, but as the author I see so many ways that it could yet be improved (outlined in proposal method in Appendix 'G'). It is hoped that we will have the opportunity to realize these aspirations. The secondary goals defined at the beginning of this project have also been substantially completed as far as possible within the restrictions of time, tools and budget. Two key insights that the author has gained in this timeframe are:

Quality data is invaluable to a quality result ("garbage in, garbage out"). In this case, much of the data was unsuitable for use in the final analysis because it was too brief to describe a trend, was missing key attributes (risk level for example), or was not permissible because it was not authorized for use. In general, the majority of this sample data will be representative of the majority of the population (patients with prostate cancer) from which it was taken. Thus the prototype built for this task is able to calculate growth probabilities for intermediate risk patients with typical PSAs based upon a reasonable number of case histories (about 30). In comparison, patient growth cases that are not typical often cannot be compared to a significant set of similar case histories because they do not happen frequently enough in a 300 patient sample. The further the patient case profile differs from the norm, the worse this problem becomes. As a result the prototype tool that we have developed does not give statistically significant results for patients with low risk, or for intermediate/high risk patients whose PSA profile is significantly elevated because they are significantly different than the usual patient. This can be corrected by significantly increasing the number of source history cases as discussed in the "Recommendations" section that follows.

Health care practitioners are not computer experts, nor should they ever need to be. Visual development tools like Borland Developer Studio function by creating a complete GUI first and then later adding the algorithms behind the GUI to make it work. This is counter intuitive to the way the real world works where we build something from pieces, so that it does not look complete until all the parts are assembled. Development of this application has been interesting at times because most computer users are not accustomed to seeing a GUI that is only a semi-functional faceplate, where portions of its coding are a temporary kluge to make it work while the full application continues to undergo revisions to complete, test and perfect. This is a common issue for developers to encounter with clients, and health care is no different.

Healthcare practitioners have some strong preconceptions about software and computer systems in general that will evolve gradually with time, patience, and experience. Health Informatics solution developers need to be equally open to change, new ideas, and unique ways of looking at problems as suggested by skilled practitioners, because only by combining our skills and knowledge effectively with one another can we achieve the best results. Solutions must “fit” the needs of the practitioners who know their subject matter best, and their patients as well. Nobody wants a “solution” that makes their goals harder to achieve.

7. Recommendations

The prototype project created during this work-term is not a development endpoint, but rather a starting point. The product application can give rough predictions of PSA growth probabilities but it is based on only the most basic of information (time since treatment, patient risk level, expected time until follow-up appointment, possible future PSA at that future date), but there is much more than can be done. The criteria have been kept very basic primarily due to the small dataset for drawing conclusions (fewer than 300 patients). An initiative is already underway with a hospital system in British Columbia to share their similar patient PSA database with CDHA and add up to 6000 more patients to the source data set. More information will give more confidence that the patterns seen in the data are characteristic of the true population behavior as opposed to irregular outlier outcomes that may give incorrect results. If this initiative, to support significant expansion of the data can be completed across provincial, administrative and procedural jurisdictions, then we will be able to continue to refine and develop this tool to a point of being a daily tool rather than an interesting prototype. As such it may lend itself to improving health outcomes as is ultimately desirable. The preliminary data request proposal to address these issues is included in this report as Appendix 'G'.

References :

- Black, A., Grubb, R., Hickey, T., Pinsky, P., Reding, D., Izmirlian, G., et al. (2009). Initial PSA levels and prostate cancer diagnosis with up to 7 years follow-up in the PLCO cancer screening trial. *The Journal of Urology*, 181 (4), 647.
- Briganti, A., Suardi, N., Gallina, A., Da Pozzo, L., Roscigno, M., Freschi, M., et al. (2009). Which patients are at real high risk for dying from prostate cancer? A long-term follow-up analysis on high risk prostate cancer patients treated in the PSA era. *The Journal of Urology*, 181 (4), 270-275.
- Capital District Health Authority, Halifax, Nova Scotia, Retrieved December 22, 2009, from Capital District Health Authority Web site: <http://www.cdha.nshealth.ca/>
- Dippon, J., Fritz, P., & Kohler, M. (2002). A statistical approach to case based reasoning, with application to breast cancer data. *Computational Statistics & Data Analysis*, 40 (3), 579-602.
- Han, M., Partin, A. W., Zahurak, M., Piantadosi, S., Epstein, J. I., & Walsh, P. C. (2003). Biochemical (prostate specific antigen) recurrence probability following radical prostatectomy for clinically localized prostate cancer. *The Journal of Urology*, 169 (2), 517-523.
- Fok, S. C., Ng, E. Y. K., & Thimm, G. L. (2003). Developing case-based reasoning for discovery of breast cancer. *Journal of Mechanics in Medicine and Biology*, 3, 231-246.
- Loughlin, K. R. (2004). Clinical guide to prostate specific antigen. Chipping Norton, Oxfordshire, Bladon Medical Publishing.
- Mayer, R (2004), Two steps forward in the treatment of colorectal cancer, *New England Journal of Medicine*, 350, 2406-2408.
- McFarlane, J., & Warren, K. (2006). PSA tracker - remote follow-up of prostate cancer patients. *European Urology Supplements*, 5 (2), 323.

McIntosh HM, Neal RD, Rose P, Watson E, Wilkinson C, Weller D, et al. (2009). Follow-up care for men with prostate cancer and the role of primary care: a systematic review of international guidelines. *British Journal of Cancer*, 100 (12), 1852-60.

Mitka, M. (2009). Guideline supports long-term use of medication to lower prostate cancer risk. *JAMA : the Journal of the American Medical Association*, 301 (17), 1753-1754.

Pantuck, A., Zomorodian, N., Rettig, M., Aronson, W., Heber, D., & Belldegrun, A. (2009). Long term follow up of phase 2 study of pomegranate juice for men with prostate cancer shows durable prolongation of PSA doubling time. *The Journal of Urology*, 181 (4), 295.

Pienta KJ (2009). Critical appraisal of prostate-specific antigen in prostate cancer screening: 20 years later, *Urology*, 73 (5 Suppl), 11–20.

Smith, J. A. (2004). Long-term follow-up of radiotherapy for prostate cancer. *Urologic Oncology*, 22 (6), 494-499.

Wilkinson, A. N., Brundage, M., & Siemens, R. (2008). Approach to primary care follow-up of patients with prostate cancer. *Canadian Family Physician*, 54, 204-210.

Zeliadt, S., Ginger, V., Etzioni, R., & Lin, D. (2009). Follow up and biopsy patterns after elevated prostate specific antigen (PSA) tests. *The Journal of Urology*, 181 (4), 751-752.

APPENDIX A – Draft Proposal

DRAFT PROPOSAL – JUNE 21, 2009

Rutledge/Wilke PROSTATE CANCER – FOLLOW UP STUDY

Purpose

- 1 – Create a software-based algorithm which would help guide the choice of follow-up intervals for individual patients treated for prostate cancer.
- 2 – Update the prostate cancer database of Dr. Rutledge, Wilke and Sun's patients allowing for outcome analysis.

Background

The PSA profile often signals recurrence of prostate cancer years in advance of a necessary clinical intervention. Patients may be followed too frequently wasting both the physician and patient time and resources. Likewise, high-risk patients seen at overly extended intervals may develop significant clinical problems between visits and miss out on the benefits of early intervention. There's little scientific literature to help guide the choice of follow up intervals according to the prognostic factors for the individual patient. A simple software program needs to be developed which could help predict the likelihood of a significant recurrence over a prescribed time. For example, the program could predict, based on the details of an individual case, the chance that the PSA will surpass 20 before a two year follow up appointment occurs. (the PSA level or length of follow up interval could be modified in this equation).

Current database

In the late 1990s Dr. Rutledge created a data base which captures the initial prognostic factors, follow up PSA results, and intervention outcome for prostate cancer treated radically at the Nova Scotia Cancer Centre. Dr. Wilke subsequently modified the database template and has continued to update the outcome for his and Dr. Sun's patients. Dr. Rutledge's patients have not been updated in years. Several hundred patients are being followed in this database. As part of this study, the question of need for rectal examination will be studied: specifically, which patients can be followed safely forgo routine rectal exams by following PSA and clinical symptoms only ie which patients could be followed safely in a telehealth environment.

Study Plan

- 1 – Identify all of Dr. Rutledge's radical prostate through the OPIS scheduling system. By reviewing all patients attending Dr. Rutledge's Wednesday afternoon GU clinic since 2000. Each male patient who

received 33 fractions or more (as verified in OPIS) before 2006 would qualify as this would guarantee 3.5 years of minimum follow up for each patient.

2 – This list of radically treated patients would be split into two groups. Those that have already been entered into the database and those that haven't. The charts of the patients who haven't been entered database will be ordered ten at a time through Dorreen, Dr. Rutledge's secretary. The initial prognostic factors and treatments fields will entered in the data base and the follow up information will be extracted as far as possible.

3 – The files of patients cross-referenced to the radically treated list will be analysed as follows. Patients outside of CDHA: PSAs will be identified in the meditech system. Patients within CDHA: PSAs will be identified using the networks database. Regardless of source the PSAs will be entered into the database. This same method of PSA extraction will be used for all qualifying patients to update the PSAs of patients not seen in clinic recently but in whom their PSA has been captured in one of the two systems.

4 – Reviewing charts of patients who appear to have relapsed. The individual charts of patients will be analyzed if the PSAs seem to be rising more than a year following end of treatment and suddenly drop to less than 20% of their peak value (implies hormone therapy). Other patients whose PSA rises to greater than 10 will have individual chart edits to review if associated clinical events occurred eg. positive bone scan, local failures etc.

5 - We will generate a list of cases in which patients developed clinical relapse between visits to capture the prognostic factors in their cases. As well DFS and OS curves will be generated for all cases.

Follow up algorithm

This part of the research project is being generated. Dr. Rutledge will approach the Informatics division of the Dalhousie Computer Science Department. The plan is to develop a computer program which will draw on the anonymous database to predict the probability of recurrence over a specified interval according to patient characteristics (eg. initial prognostic factors, current PSA level, PSA doubling time if PSA rising, number of years in follow up etc).

Budget

For now, there are no associated costs to this proposal. Whether a small stipend should be paid for data entry or for computer science input is yet to be determined. A second draft of this proposal will follow when more information is available.

APPENDIX B – Activity Summary

Date	Start	Duration	Activity
14-Jul-09	10:45	2	Dr Rutledge Project Intro/Overview
14-Jul-09	19:00	1	Search for Follow Up Articles
15-Jul-09	13:15	3	Meet staff, participants, see charts
15-Jul-09	18:00	1	Draft and submit internship forms to Grace
16-Jul-09	13:30	1	Work Term Reg HINF 7000, Scan Data from Dr Wilke
16-Jul-09	18:30	2.5	Collect 'R' Stats Tools, Convert data to CSV
17-Jul-09	8:00	4	Review 2 articles, Kellogg Books, Work Space
18-Jul-09	13:00	0.5	Read & Respond emails
19-Jul-09	19:00	2	Parse out data for Dr Wilke patients
20-Jul-09	8:00	4	Filter Data, map PSA and time in Julian only
20-Jul-09	18:30	1.5	Exploratory plots, refine data, note data quality issues
21-Jul-09	8:00	4	Meeting with Ron Dewar (CCNS), Epidemiologist, Forms
22-Jul-09	10:00	4	Find follow-up articles & book "Clinical guide to PSA"
23-Jul-09	9:00	4	Prioritize data fields based for MLRegression & R/Excel
24-Jul-09	13:00	3	Review articles and summarize to Dr Rutledge
27-Jul-09	8:30	4	Write 'R' program to move patient treatment dates, T score and Gleason score to PSA Data for only Dr Wilke's patients.
28-Jul-09	10:00	1	Meeting to discuss detailed project requirements
31-Jul-09	12:00	2	Identify missing numbers of patient data records
03-Aug-09	13:00	2	Refine data matching from PSA to Patient Data and vice versa
04-Aug-09	9:00	2	Add in patient data marked as DW and recalculate PSA data
07-Aug-09	22:00	1	Explore frequency of PSA results per patient
11-Aug-09	12:00	4	Isolate patients missing numerous critical data fields
12-Aug-09	10:00	6	Review OPIS data, Meeting with Dr Rutledge. Parse 'R'.
13-Aug-09	8:00	5	Read disease prediction methods in 'R'. Write more code.
14-Aug-09	7:30	4	Use treatment date PSA values as predictor, not resultant
17-Aug-09	8:00	4	Correlation to time since procedure is non-existent. Why?
18-Aug-09	8:00	4	Find data supporting hypothesis (PSA escalates with time)
19-Aug-09	8:00	3	Read epidemiology modeling in plants using R for prediction
20-Aug-09	8:00	4	Produced initial graphs and regression estimates of cases
20-Aug-09	18:30	5	Generalize models to produce a more generic formula
21-Aug-09	8:00	3	Research 'R' methods, especially multivariate regression modeling
22-Aug-09	18:00	4	LGM models fail, looking for a higher fidelity solution.
23-Aug-09	18:00	6	Experiment 3rd, 5th, and 7th degree polynomial models.
24-Aug-09	9:00	4	Summarize issues and question for discussion
25-Aug-09	8:00	4	Rework clipping criteria, talk about results of plots
25-Aug-09	13:00	4	Nearest neighbor in a 3D array to store case results? How? Dr Rutledge prefers unmodified patient history plots.
25-Aug-09	21:00	3	Look for a free C or C++ compiler, GCC too big and awkward
26-Aug-09	8:00	4	Present work for comment, look for an R compiler (is none)
26-Aug-09	13:00	4	Begin rewrite to use straight plots vice regression models
27-Aug-09	8:00	4	Reconsider data exclusion criteria, continue algorithm rewrite
27-Aug-09	13:00	5	Numerous problems with computing start and end window
28-Aug-09	8:00	4	Count all graphs greater than threshold at end, Trial 0.2 minimum bin size
30-Aug-09	18:00	6	Experiment with ways to integrate R to C++ for construction of a final product. Look for ways to make composite graphs
31-Aug-09	7:30	4	Find patients with many valid PSA data points, but cannot be used because they miss a few pieces of other data
31-Aug-09	18:00	4	Search for a good IDE that fits the budget (\$0). Top contenders are JAVA&NetBeans or TurboCPP2006

01-Sep-09	8:00	4	Try TurboCPP2006. It is old and hard to find the correct drivers
02-Sep-09	8:00	8	Move to resident's room. Desk by window. Set to relearn Borland Builder instead of TurboCPP (it's been a few years).
03-Sep-09	7:30	6	Build PSA Calculator GUI with Borland Builder. I'm not happy with the Date combo boxes, they require too much user input
04-Sep-09	8:00	8	Continue on GUI, rework R code to produce chart per patient
05-Sep-09	12:00	6	Replace combo boxes with DateTimePickers, find a bug in R that always crashes processing on chart 121611. find a variable typing mismatch that the interpreter did not flag
06-Sep-09	9:00	4	Parameter passing working from C++ to R. input graphs from R to C++. Need to get numeric result from R to C++
07-Sep-09	11:00	4	Add *Open* ability to look at individual chart graphs
08-Sep-09	8:00	6	Add inline documentation while I still remember the code well
09-Sep-09	8:00	6	More online documentation in the R code, and fix a bug which caused the extrapolated line to be drawn as well as the grey
10-Sep-09	8:00	4	Get 20 pretreatment PSAs from 20 charts, some cases are not relevant because they has a prostatectomy prior to RT
11-Sep-09	8:00	3.5	Dr Rutledge wants a probability over time for the threshold
14-Sep-09	8:00	6.5	Implement the probability over time graphing for threshold
15-Sep-09	10:00	5	The R scripts that I have been using are showing the signs of multiple adds, changes, deletes, cludges, etc I am going to rewrite them using standardized constants
16-Sep-09	10:00	5	Add "term view" and "future date view" radio buttons to give the user a choice of display type
17-Sep-09	10:00	5	Explain to Dr Rutledge why I am making assumptions in my clipping algorithm. We find there is more data than I have been given that has the dates for follow-up interventions ...
18-Sep-09	9:00	5	Dr Rutledge suggest that Dr Wilke does have the data and asked him to forward that data to me, after searching all date fields
21-Sep-09	8:00	6.5	Improve user prompting and context sensitive help on main C++ interface. Still cleaning up R scripts ...
22-Sep-09	10:00	4.5	Resolve some graphing errors added by new constants
23-Sep-09	8:00	6	Cleaning up RfromC_longview and shortview. Working fairly well now
24-Sep-09	8:00	4	Fix fonts, hints, GUI, make sure Y axis decimal point is readable so not to confuse 2.1 and 21 (example)
25-Sep-09	13:00	3	Give detailed status of project and outline of steps required to complete it. IE. Prototype over expectation
28-Sep-09	8:00	6.5	Parse AD Intervention data file, merge it with the patient data file, rework the algorithm for charting and prediction
29-Sep-09	8:00	5	The new patient and PSA data files are not working, and always result in only 1 chart being examined. Why?
30-Sep-09	8:00	4.5	Strip down both the patient and PSA data files to be the minimum data set needed (about 6 columns each). Rmerge patient file
01-Oct-09	8:00	5	Rewrite the data prep routine and get new and minimized prep files. Rewrite RfromCLong and RfromCShort. Works !
05-Oct-09	8:00	6.5	Reorganize the C++ GUI to move the view buttons and add facility to display the list of exceeding chart numbers
06-Oct-09	8:00	5	Find a bug that always takes the fileopendialog back to the last folder that was opened ... install latest version on Rob's machine
07-Oct-09	9:00	5.5	Added a new timer method to time the opening of the R scripts, not happy with the result so I will put that on the backburner for now, meet with Dave MacFarland and Rob at 2pm, demo tool and find another bug (of course)
08-Oct-09	10	5	Track bug to an unclosed file pointer in the file open method
12-Oct-09	8:00	4	Thanksgiving holiday

13-Oct-09	10:00	4.5	Fix the patient chart file pointer and update the chart generation script to add red segments for failed cases
14-Oct-09	8:00	4.5	Add red graphs for failed cases, resize the GUI, Have tour of the new emerg with John Baker. Research expo in the Bethune
15-Oct-09	10:00	4.5	Make a release build, retest, show Dr Rutledge results can be fickle, reducing initial PSA = generate results with more failures
16-Oct-09	13:00	2.5	Review files formats and baseline scripts. Contact Lillian Coshell at Cobequid to review their systems for data processing info
19-Oct-09	9:00	5	Develop user manual framework offsite (nasty head cold).
20-Oct-09	10:00	4.5	Make screenshots from all the tools that were used in this project
21-Oct-09	12:00	4.5	More screenshots and resizing, investigated bug that makes graphs appear 1 month off and found a rounding error in MonthInDays
22-Oct-09	10:00	4.5	Update the data prep steps stored in the data file folder
26-Oct-09	8:00	8	Completed first draft of user manual
27-Oct-09	8:00	4.5	Separate final executables, data files, etc for delivery CD
28-Oct-09	8:00	4.5	Reformat data into a CSV compatible with WEKA
29-Oct-09	12:00	1.5	Emails, Doreen, schedule meetings, user doc
30-Oct-09	13:00	3.5	user doc
02-Nov-09	8:00	6.5	Import patient data to WEKA ... many problems doing this with unexpected characters in the files, unequal lines lengths, etc
03-Nov-09	9:00	5	Get WEKA data working, convert to ARFF and reclassify as low/med/hi risk levels, send chart to Dr Rutledge. Discussion with Raza about process flow analysis
04-Nov-09	9:00	2	emails about process flow analysis to Allan Day, Dr Rutledge, Raza
05-Nov-09	9:00	2	emails with Paul Oliver,
09-Nov-09	8:00	7	User guide - How to prep data
10-Nov-09	8:00	4	User guide- How to prep data
11-Nov-09		0	H1N1 - cough, gasp, wheeze
12-Nov-09		0	H1N1 - cough, gasp, wheeze
13-Nov-09		0	H1N1 - cough, gasp, wheeze
14-Nov-09		0	H1N1 - cough, gasp, wheeze
15-Nov-09		0	H1N1 - cough, gasp, wheeze
16-Nov-09		0	H1N1 - cough, gasp, wheeze
17-Nov-09		0	H1N1 - cough, gasp, wheeze
18-Nov-09	8:00	5.5	Meeting with department doctors, Paul Oliver, Carol Anne, others. Create rough slides, guide, outline email for Dr Rutledge to BC
19-Nov-09	10:00	5	Review time guarantee planning level 1, 2, and 3 process maps. Send summary to Raza. PSA tool to Graham Cook. Sia is sick (uhoh!)
20-Nov-09	15:00	4.5	Helped with Capital Health H1N1 Immunization Clinic at the Sackville Sportsplex. Collect/organize patient consent forms from RNs, filed by 1-shot or 2-shot. Direct patients. Catered supper
23-Nov-09	8:00	6.5	Try a number of different data mining steps on the patient data, showed the results to Graham Cook, mail copy to Dr Rutledge
24-Nov-09	8:00	4.5	Met with Dr Rutledge and Graham at 11AM, we discuss our model assumptions that we need to examine further
25-Nov-09	17:00	3.5	Change level medium to intermediate, create more screenshot jpgs for Dr Rutledge's presentation, send articles to Graham Cook
30-Nov-09	8:00	6	Review more articles, those in 2009 only whenever possible
			>> Slow down hours so they last until we can finish data requests (vacations and holidays start slowing things down) <<
09-Dec-09	10:00	6	First draft of expanded patient data transfer from Tom Pickles
11-Dec-09	10:00	4	Second draft of expanded patient data transfer from Tom Pickles - add new intro from Dr Rutledge (to be revised by Graham Cook)
22-Dec-09	8:00	6.5	Moved all source code and documentation from laptop to desktop, reinstalled

			C++, reinstall R, recompiled code, seek revisions from Graham (still not available). Created guide/ report skeleton
26-Dec-09	12:00	4	Updated user guide, created PDF, created final delivery disc
27-Dec-09	12:00	3.5	Printer user manual in color, revised data request for Tom Pickles
29-Dec-09	12:00	4	Get feedback from Graham, rewrite the first 1/3 of the project overview and request. Send it back to Graham again for comments.
31-Dec-09		455	Total Hours. Will continue on volunteer basis.

APPENDIX C – ‘R’ Script to create “Future Date View”

```

#####
#####
## Title: RfromC_shortview.R
## Author: Tom Harding DAL B00120003
## Date : 21 Sept 2009
##
## This script is called by the PSA Calculator application to conduct
## the data manipulation and graphing required to determine the past
## behavior of patient PSA trajectories which parallel a current
## case and give a report on what past recorded experience indicates
## for 1 specific date in the future
##
## NB: At this time the database of recorded cases is not very large
## and in many cases may yield a result based upon 5 - 10 (or fewer)
## cases. This is not a statistically large number and strict conclusions
## should not be drawn from these results alone.
##
#####
#####

## DEFINE CONSTANTS indicating column positions for data in files and matrices
Input_RISK = 1
Input_STARTdate = 2
Input_START_PSA = 3
Input_ENDdate = 4
Input_END_PSA = 5

PatRawData_CHART = 1
PatRawData_RISK = 2
PatRawData_RTdate = 8
PatRawData_INTdate = 13
PatRawData_MINPOINT = 14      # point minimum was calculated in first script
PatRawData_MAXPOINT = 15      # point maximum was calculated in first script
PatRawData_NUMPOINTS = 16     # point count was calculated in first script

PatPSADData_CHART = 1
PatPSADData_PSA = 7
PatPSADData_TestDate = 6
PatPSADData_RTdate = 8        # where we add RT dates to the PSA data
PatPSADData_FailDate = 9      # where we add PSA Fail Date

ProtoPatData_CHART = 1
ProtoPatData_RISK = 2
ProtoPatData_RTdate = 8
ProtoPatData_INTdate = 13
ProtoPatData_MINPOINT = 14    # point minimum was calculated in first script
ProtoPatData_MAXPOINT = 15    # point maximum was calculated in first script
ProtoPatData_NUMPOINTS = 16   # point count was calculated in first script

ProtoPSADData_CHART = 1
ProtoPSADData_PSA = 7
ProtoPSADData_DATE = 6

Bucket_CHART = 1
Bucket_PreScore = 2
Bucket_PostScore = 3
Bucket_NowScore = 4

LINE_PSA_Data_CHART = 1
LINE_PSA_Data_DATE = 2
LINE_PSA_Data_PSA = 3

MonthInDays = 30.4375          # (365*4+1)/48

##### Read The Calculation Parameters From C++ #####
##
## read the parameter file created by the C++ GUI ##
Rparams <-read.csv(file="Rparams.csv",head=FALSE,sep=",")
##
## set the risk group based upon the value in the input file
LowMedHigh = as.integer(Rparams[Input_RISK])
##
## set the Start Window and End Window dates based upon the values in the input file
START = as.integer(Rparams[Input_STARTdate])
END = as.integer(Rparams[Input_ENDdate])
##
## set the Start PSA and End PSA based upon the values in the input file
STARTV = as.numeric(Rparams[Input_START_PSA])
ENDV = as.numeric(Rparams[Input_END_PSA])

```

```

##
## read the patient and PSA data that has been already "prepped" by a prep script
## see also the excel prep steps in "keepersteps.txt"
OurPats_PSA_Data <- read.csv(file="Prepped OurPats_PSA_Data.csv",head=TRUE,sep=",")
PatRawData <- read.csv(file="Prepped PatRawData.csv",head=TRUE,sep=",")
##
## low risk indicated
if (0 == LowMedHigh)
{
  PatRawData = PatRawData[PatRawData[,PatRawData_RISK]=="low",]
  OurPats_PSA_Data = OurPats_PSA_Data[OurPats_PSA_Data[,PatPSAData_CHART] %in% PatRawData[,PatRawData_CHART],]
}
##
## medium risk indicated
if (1 == LowMedHigh)
{
  PatRawData = PatRawData[PatRawData[,PatRawData_RISK]=="intermediate",]
  OurPats_PSA_Data = OurPats_PSA_Data[OurPats_PSA_Data[,PatPSAData_CHART] %in% PatRawData[,PatRawData_CHART],]
}
##
## high risk indicated
if (2 == LowMedHigh)
{
  PatRawData = PatRawData[PatRawData[,PatRawData_RISK]=="high",]
  OurPats_PSA_Data = OurPats_PSA_Data[OurPats_PSA_Data[,PatPSAData_CHART] %in% PatRawData[,PatRawData_CHART],]
}
##
##### Done Reading The Calculation Parameters From C++ #####

{
  if (END > START)
  {
##### Calc the first window (bucket) min and max #####
##
## the first window is +/- 50% by default
topbuffer = 0.0 # ensure it is a float variable
topbuffer = STARTV*1.5
bottombuffer = 0.0 # ensure it is a float variable
bottombuffer = STARTV*0.5
##
## window is never smaller than +/- 0.20
if (topbuffer < STARTV+0.2)
  topbuffer = STARTV+0.2
if (bottombuffer > STARTV-0.2)
  bottombuffer = STARTV-0.2
if (bottombuffer < 0)
  bottombuffer = 0
##
## create float result variables
Result = 0.0
Denominator = 0.0
Numerator = 0.0
##
##### Calc the first window (bucket) min and max #####

#####Individual Plots and Start Bucket#####
##
## make a working copy of the patient and PSA data
ProtoPatData = PatRawData[PatRawData[,PatRawData_NUMPOINTS] > 4,] # we have at least 4 points for this patient
ProtoPSAData = OurPats_PSA_Data [OurPats_PSA_Data[,PatPSAData_CHART] %in% (ProtoPatData[,ProtoPatData_CHART]),]
##
## makes a more streamlined version of the PSA data with only Chart#/Date/PSA
LINE_PSA_Data = cbind(ProtoPSAData[,ProtoPSAData_CHART], ProtoPSAData[,ProtoPSAData_DATE], ProtoPSAData[,ProtoPSAData_PSA])
##
## make a start bucket scratchpad matrix with only Chart#/PSA-left/PSA-right/PSA-now/fit_indicator
LINE_Start_Bucket = cbind(ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1])
##
## set the PSA-left/PSA-right/PSA-now/fit_indicator fields to -1 meaning "no statement"
LINE_Start_Bucket[,2:4] = -1
##
#####

#####FIND Charts that match START parameters #####
## continue processing if we have at least 1 patient to consider
## process if we have at least 1 patient to consider
if (0<length(ProtoPatData[,ProtoPatData_CHART]))
{

```

```

## look through the list of patients one by one
for (i in 1: length(ProtoPatData[,ProtoPatData_CHART]))
{
  ## get the chart number for this patient
  ThisChartNum = ProtoPatData[i,ProtoPatData_CHART]
  ## get all the PSA plot entries for this patient
  ThisChartNum_LINE_PSA_Data = LINE_PSA_Data[LINE_PSA_Data[,LINE_PSA_Data_CHART]==ThisChartNum,]

  # sort by date
  ThisChartNum_LINE_PSA_Data = ThisChartNum_LINE_PSA_Data[order(as.numeric(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE])),]

  # this section of the script creates a plot of the PSA values for each patient that is examined
  # these can be opened from the "File"/"Open" command on the PSA Calculator if there is a particular patient of interest

  win.metafile( (paste("chartplot", ThisChartNum, ".wmf"))) # prepare plot device

  par(font.axis=10)
  plot (as.integer((ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA],
main=paste("Plot For Chart:",ThisChartNum), col="red", type = "p", xlab="Months Since RT", ylab="PSA")

  k=length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
  ThisChartNum_LINE_PSA_Data = rbind (ThisChartNum_LINE_PSA_Data, ThisChartNum_LINE_PSA_Data[k,])
  k=k+1
  ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE] = 10 * 365 # maximum date used here

  if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate]) )
  {
    ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = 100
  }
  else
  {
    ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA]
  }

  points (as.integer((ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA],
main=ThisChartNum, col="blue", type = "l")

  if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate]) )
  {
    points ( x = c( as.integer((ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),
as.integer((ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays)),
y = c( ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA], ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA]),
col="red", type = "l")
  }

  # close the output file
  dev.off()

  # examine the PSA values of all the points we kept
  NumPoints = length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
  # until we have looked at them all or found the before/after gap we were looking for
  DONE=FALSE

  for (j in 1:NumPoints)
  {
    # if the plot starts before the date of the start window
    if (ThisChartNum_LINE_PSA_Data[1,LINE_PSA_Data_DATE] < START)
    {
      # if we find a PSA that is to the right of the start window date
      if (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] > START && !DONE)
      {
        LINE_Start_Bucket[i,Bucket_PreScore] = ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_PSA] # psa before start date
        LINE_Start_Bucket[i,Bucket_PostScore] = ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_PSA] # psa after start date

        DY = 1.0 * (LINE_Start_Bucket[i,Bucket_PostScore] - LINE_Start_Bucket[i,Bucket_PreScore]) # after psa - before psa
        DX = 1.0 * (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] - ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_DATE]) # after time - before
time
        LINE_Start_Bucket[i,Bucket_NowScore] = LINE_Start_Bucket[i,Bucket_PreScore] + DY/DX * as.integer((START - ThisChartNum_LINE_PSA_Data[j-
1,LINE_PSA_Data_DATE]))
        DONE=TRUE
      }
    }
  }

  # find the values above the minimum
  LINE_Start_Bucket_low= rbind(LINE_Start_Bucket[LINE_Start_Bucket[,Bucket_NowScore] < topbuffer , ])
  # find the values below the maximum
  LINE_Start_Bucket_high= rbind(LINE_Start_Bucket[LINE_Start_Bucket[,Bucket_NowScore] > bottombuffer , ])
  # fine the value that are in both of these groups
  LINE_Start_Bucket = rbind(LINE_Start_Bucket_low [LINE_Start_Bucket_low[,Bucket_CHART] %in% (LINE_Start_Bucket_high[,Bucket_CHART]),])
}
#####DONE :FIND Charts that match START parameters #####

```

```
##### Find Charts Matching Future Parameter #####
##
## if we have at least one patient who matched the start condition
##
if (0<length(LINE_Start_Bucket[,Bucket_CHART]))
{
  ## find the patient data for all those charts that matched the start parameter (+/- 50%)
  ProtoPatData = ProtoPatData [ProtoPatData[,ProtoPatData_CHART] %in% (LINE_Start_Bucket[,Bucket_CHART]),]
  # find the PSA data for all those charts that matched the start parameter (+/- 50%)
  ProtoPSAData = ProtoPSAData [ProtoPSAData[,ProtoPSAData_CHART] %in% (ProtoPatData[,ProtoPatData_CHART]),]

  ## makes a more streamlined version of the PSA data with only Chart#/Date/PSA
  LINE_PSA_Data = cbind(ProtoPSAData[,ProtoPSAData_CHART], ProtoPSAData[,ProtoPSAData_DATE], ProtoPSAData[,ProtoPSAData_PSA])

  ## make a start bucket scratchpad matrix with only Chart#/PSA-left/PSA-right/PSA-now/Eit_indicator
  LINE_End_Bucket = cbind(ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1])
  LINE_End_Bucket[,2:4] = -1

  # examine each patient that crossed the first criteria, to see if they exceed the second criteria
  for (i in 1: length(ProtoPatData[,ProtoPatData_CHART]))
  {
    # get the chartnumber for next patient to examine
    ThisChartNum = ProtoPatData[i,ProtoPatData_CHART]
    # get the PSA data for this patient only
    ThisChartNum_LINE_PSA_Data = LINE_PSA_Data[LINE_PSA_Data[,LINE_PSA_Data_CHART]==ThisChartNum,]

    # sort PSA points by date
    ThisChartNum_LINE_PSA_Data = ThisChartNum_LINE_PSA_Data[order(as.numeric(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE])),]

    k=length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
    ThisChartNum_LINE_PSA_Data = rbind (ThisChartNum_LINE_PSA_Data, ThisChartNum_LINE_PSA_Data[k,])
    k=k+1
    ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE] = 10 * 365 # maximum date used here
    if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate] ) )
    {
      ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = 100
    }
    else
    {
      ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA]
    }
    #how many points are we keeping?
    NumPoints = length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])

    DONE=FALSE
    # until done, look through all the points and estimate the value of the chart at the precise day of interest
    for (j in 1:NumPoints)
    {
      if (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] > END && !DONE)
      {
        if (j > 1)
        {
          LINE_End_Bucket[i,Bucket_PreScore] = ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_PSA] # psa before ThisDate
          LINE_End_Bucket[i,Bucket_PostScore] = ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_PSA] # psa after ThisDate

          DY = 1.0 * (LINE_End_Bucket[i,Bucket_PostScore] - LINE_End_Bucket[i,Bucket_PreScore]) # after psa - before psa
          DX = 1.0 * (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] - ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_DATE]) # after time - before
time
          LINE_End_Bucket[i,Bucket_NowScore] = LINE_End_Bucket[i,Bucket_PreScore] + DY/DX * as.integer((END - ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_DATE]))
        }
        DONE=TRUE
      }
    }

    # we only want to keep those points that exceeded the end value selected
    LINE_End_Bucket_over= rbind(LINE_End_Bucket[LINE_End_Bucket[,Bucket_NowScore] > ENDV , ]) # must rbind to keep 2 dimensions

##### how many exceeded the end value % #####
{
  Denominator = length(LINE_Start_Bucket[,Bucket_CHART])
  Numerator = length(LINE_End_Bucket_over[,Bucket_CHART])

  if (0 == Denominator)
  {
    Result = -1
  }
  else
  {
    Result = (Numerator) / (Denominator * 1.0)
  }
}
}

#####Save output results for C++ interface #####
{
  sink("Cparams.txt")
}
```

```

print("LINE_End_Bucket_over")
print(length(LINE_End_Bucket_over[,Bucket_CHART]))
print("LINE_Start_Bucket")
print(length(LINE_Start_Bucket[,Bucket_CHART]))

if(0 < length(LINE_End_Bucket_over[,Bucket_CHART]))
{
  for (i in 1: length(LINE_End_Bucket_over[,Bucket_CHART]))
  {
    print(as.numeric(LINE_End_Bucket_over[i,Bucket_CHART]))
  }
}
sink()
}
#####

##### PLOT graphs in grey and in color #####
# if there was at least one graph that crossed the start window
if (0<length(LINE_Start_Bucket[,Bucket_CHART]))
{
  # size the output window by setting it up to plot a dummy (invis) graph
  win.metafile( paste("chartexceeded.wmf")) # prepare plot device
  dummyy = c (0,ENDV * 3.5)
  END_Graph = END * 1.25
  if(END_Graph > 3650) END_Graph = 3650 #never plot past 10 years
  dummyx = c (0,END_Graph)
  par(font.axis=10) # set font for graph axis to be more legible
  plot(as.integer(dummyx/MonthInDays),dummyy, main=(paste(as.integer(Result*10000)/100.0,"% Exceeded At Future Date")), col = "green", type = "n",
  xlab="Months Since RT", ylab="PSA")

  # look at all the charts that exceeded the end value
  for (i in 1: length(LINE_End_Bucket[,Bucket_CHART]))
  {
    # get the chart number of this graph
    ThisChartNum = LINE_End_Bucket[i,Bucket_CHART]
    # get the PSA values for this graph
    ThisChartNum_LINE_PSA_Data = LINE_PSA_Data[LINE_PSA_Data[,LINE_PSA_Data_CHART]==ThisChartNum,]

    # sort PSA plot points by date
    ThisChartNum_LINE_PSA_Data = ThisChartNum_LINE_PSA_Data[order(as.numeric(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE])),]

    # if there is at least 1 point in this PSA dataset then plot all its POINTS in GREY
    if (0 < length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]))
      points (as.integer((ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA],
      col="grey", type = "p")
      k=length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
      ThisChartNum_LINE_PSA_Data = rbind (ThisChartNum_LINE_PSA_Data, ThisChartNum_LINE_PSA_Data[k,])
      k=k+1
      ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE] = 10 * 365 # maximum date used here
      if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate] ) )
      {
        ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = 100
      }
      else
      {
        ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA]
      }
    }
    # then plot this graphs LINES in GREY
    if (0 < length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]))
      points (as.integer((ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA],
      col="grey", type = "l")
  }

  # if there was at least one case that exceeded the end value
  # then plot it in color
  if (0<length(LINE_End_Bucket_over[,Bucket_CHART]))
  {
    dummyy = c (ENDV, 100)
    dummyx = c (END, END)
    points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, main="Exceeding Charts", col = "black", type = "l")
    dummyy = c (ENDV, 100)
    dummyx = c (END, END)
    points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, col = "black", type = "p")
    dummyy = c (bottombuffer, topbuffer)
    dummyx = c (START, START)
    points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, col = "black", type = "l")
    dummyy = c (bottombuffer, topbuffer)
    dummyx = c (START, START)
    points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, col = "black", type = "p")

    # look at each case that exceeded the end
    for (i in 1: length(LINE_End_Bucket_over[,Bucket_CHART]))
    {
      # get the chart number of this case to plot

```

```

ThisChartNum = LINE_End_Bucket_over[i,Bucket_CHART]
# get the PSA values for this case to plot
ThisChartNum_LINE_PSA_Data = LINE_PSA_Data[LINE_PSA_Data[,LINE_PSA_Data_CHART]==ThisChartNum,]

# sort the PSA value by date
ThisChartNum_LINE_PSA_Data = ThisChartNum_LINE_PSA_Data[order(as.numeric(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE])),]

# if this chart has more than one data point, then plot the POINT in RED
if (0 < length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]))
  points (as.integer((ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA],
col="red", type = "p")

k=length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
ThisChartNum_LINE_PSA_Data = rbind (ThisChartNum_LINE_PSA_Data, ThisChartNum_LINE_PSA_Data[k,])
k=k+1
ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE] = 10 * 365 # maximum date used here
if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate] ) )
  {
    ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = 100
  }
else
  {
    ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA]
  }

# if there is at least 1 point to plot, then plot the graph LINES in BLUE
if (0 < length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]))
  {
    points (as.integer((ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA],
col="blue", type = "l")

    if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate] ) )
      {
        #points ( x = c( as.integer((ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),
as.integer((ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays)),
        # y = c( ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA], ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA]),
        # col="red", type = "l" )

        points
(as.integer((ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE]+MonthInDays/2)/MonthInDays),ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA], col="red", type =
"l")
      }
    }
  }
}
else
{
# just add the start and end window bars to the graph
dummyy = c (ENDV, 100)
dummyx = c (END, END)
points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, main="Exceeding Charts", col = "black", type = "l")
dummyy = c (ENDV, 100)
dummyx = c (END, END)
points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, main="Exceeding Charts", col = "black", type = "p")
dummyy = c (bottombuffer, topbuffer)
dummyx = c (START, START)
points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, col = "black", type = "l")
dummyy = c (bottombuffer, topbuffer)
dummyx = c (START, START)
points(as.integer((dummyx+MonthInDays/2)/MonthInDays),dummyy, col = "black", type = "p")
}
}
} ## if end > start
else
{
win.metafile( (paste("chartexceeded.wmf"))) # prepare plot device
plot(100,100, main="ERROR:End Date Not After Start Date", col = "blue", type = "n", xlab="Days Since RT", ylab="%")
}
}

# if a plotting device (file) is open, then close it
if (1<dev.cur())
dev.off()

```

APPENDIX D – ‘R’ Script to create “Term View”

```
#####
## Title: RfromC_longview.R
## Author: Tom Harding DAL B00120003
## Date : 8 Sept 2009
##
## This script is called by the PSA Calculator application to conduct
## the data manipulation and graphing required to determine the past
## behavior of patient PSA trajectories which parallel a current
## case and give a report on what past recorded experience indicates
## would be the percentage chance of exceeding that value over time
##
## NB: At this time the database of recorded cases is not very large
## and in many cases may yield a result based upon 5 - 10 (or fewer)
## cases. This is not a statistically large number and strict conclusions
## should not be drawn from these results alone.
##
#####

## DEFINE CONSTANTS indicating column positions for data in files and matrices
Input_RISK = 1
Input_STARTdate = 2
Input_START_PSA = 3
Input_ENDdate = 4
Input_END_PSA = 5

PatRawData_CHART = 1
PatRawData_RISK = 2
PatRawData_RTdate = 8
PatRawData_INTdate = 13
PatRawData_MINPOINT = 14      # point minimum was calculated in first script
PatRawData_MAXPOINT = 15      # point maximum was calculated in first script
PatRawData_NUMPOINTS = 16     # point count was calculated in first script

PatPSAData_CHART = 1
PatPSAData_PSA = 7
PatPSAData_TestDate = 6
PatPSAData_RTdate = 8         # where we add RT dates to the PSA data
PatPSAData_FailDate = 9      # where we add PSA Fail Date

ProtoPatData_CHART = 1
ProtoPatData_RISK = 2
ProtoPatData_RTdate = 8
ProtoPatData_INTdate = 13
ProtoPatData_MINPOINT = 14    # point minimum was calculated in first script
ProtoPatData_MAXPOINT = 15    # point maximum was calculated in first script
ProtoPatData_NUMPOINTS = 16   # point count was calculated in first script

ProtoPSAData_CHART = 1
ProtoPSAData_PSA = 7
ProtoPSAData_DATE = 6

Bucket_CHART = 1
Bucket_PreScore = 2
Bucket_PostScore = 3
Bucket_NowScore = 4

LINE_PSA_Data_CHART = 1
LINE_PSA_Data_DATE = 2
LINE_PSA_Data_PSA = 3

MonthInDays = 30.4375         # (365*4+1)/48

##### Read The Calculation Parameters From C++ #####
##
## read the parameter file created by the C++ GUI ##
Rparams <- read.csv(file="Rparams.csv",head=FALSE,sep=",")
##
## set the risk group based upon the value in the input file
LowMedHigh = as.integer(Rparams[Input_RISK])
##
## set the Start Window and End Window dates based upon the values in the input file
START = as.integer(Rparams[Input_STARTdate])
END = as.integer(Rparams[Input_ENDdate])
DATESTEP = 30
##
## set the Start PSA and End PSA based upon the values in the input file
STARTV = as.numeric(Rparams[Input_START_PSA])
ENDV = as.numeric(Rparams[Input_END_PSA])
##
## read the patient and PSA data that has been already "prepped" by a prep script
## see also the excel prep steps in "keepersteps.txt"
OurPats_PSA_Data <- read.csv(file="Prepped OurPats_PSA_Data.csv",head=TRUE,sep=",")
PatRawData <- read.csv(file="Prepped PatRawData.csv",head=TRUE,sep=",")
##
```



```

## low risk indicated
if (0 == LowMedHigh)
{
  PatRawData = PatRawData[PatRawData[,PatRawData_RISK]=="low",]
  OurPats_PSA_Data = OurPats_PSA_Data[OurPats_PSA_Data[,PatPSAData_CHART] %in% PatRawData[,PatRawData_CHART],]
}
##
## medium risk indicated
if (1 == LowMedHigh)
{
  PatRawData = PatRawData[PatRawData[,PatRawData_RISK]=="intermediate",]
  OurPats_PSA_Data = OurPats_PSA_Data[OurPats_PSA_Data[,PatPSAData_CHART] %in% PatRawData[,PatRawData_CHART],]
}
##
## high risk indicated
if (2 == LowMedHigh)
{
  PatRawData = PatRawData[PatRawData[,PatRawData_RISK]=="high",]
  OurPats_PSA_Data = OurPats_PSA_Data[OurPats_PSA_Data[,PatPSAData_CHART] %in% PatRawData[,PatRawData_CHART],]
}
##
##### Done Reading The Calculation Parameters From C++ #####

{
if (END > START)
{
##### Calc the first window (bucket) min and max #####
##
## the first window is +/- 50% by default
topbuffer = 0.0      # ensure it is a float variable
topbuffer = STARTV*1.5
bottombuffer = 0.0  # ensure it is a float variable
bottombuffer = STARTV*0.5
##
## window is never smaller than +/- 0.20
if (topbuffer < STARTV+0.2)
  topbuffer = STARTV+0.2
if (bottombuffer > STARTV-0.2)
  bottombuffer = STARTV-0.2
if (bottombuffer < 0)
  bottombuffer = 0
##
## create float result variables
Result = 0.0
Denominator = 0.0
Numerator = 0.0
##
##### Calc the first window (bucket) min and max #####

#####Individual Plots and Start Bucket#####
##
## make a working copy of the patient and PSA data
ProtoPatData = PatRawData[PatRawData[,PatRawData_NUMPOINTS] > 4,] # we have at least 4 points for this patient
ProtoPSAData = OurPats_PSA_Data [OurPats_PSA_Data[,PatPSAData_CHART] %in% (ProtoPatData[,ProtoPatData_CHART]),]
##
## makes a more streamlined version of the PSA data with only Chart#/Date/PSA
LINE_PSA_Data = cbind(ProtoPSAData[,ProtoPSAData_CHART], ProtoPSAData[,ProtoPSAData_DATE], ProtoPSAData[,ProtoPSAData_PSA])
##
## make a start bucket scratchpad matrix with only Chart#/PSA-left/PSA-right/PSA-now/fit_indicator
LINE_Start_Bucket = cbind(ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1])
##
## set the PSA-left/PSA-right/PSA-now/fit_indicator fields to -1 meaning "no statement"
LINE_Start_Bucket[,2:4] = -1
##
#####

#####FIND Charts that match START parameters #####
## continue processing if we have at least 1 patient to consider
## process if we have at least 1 patient to consider
if (0<length(ProtoPatData[,ProtoPatData_CHART]))
{
  ## look through the list of patients one by one
  for (i in 1: length(ProtoPatData[,ProtoPatData_CHART]))
  {
    ## get the chart number for this patient
    ThisChartNum = ProtoPatData[i,ProtoPatData_CHART]
    ## get all the PSA plot entries for this patient
    ThisChartNum_LINE_PSA_Data = LINE_PSA_Data[LINE_PSA_Data[,LINE_PSA_Data_CHART]==ThisChartNum,]

    # sort by date
    ThisChartNum_LINE_PSA_Data = ThisChartNum_LINE_PSA_Data[order(as.numeric(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE])),]

    # this section of the script creates a plot of the PSA values for each patient that is examined
  }
}
}
}

```

```

# these can be opened from the "File"/"Open" command on the PSA Calculator if there is a particular patient of interest
### win.metafile( (paste("chartplot", ThisChartNum, ".wmf")) ) # prepare plot device
### plot (ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE],ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA], main=ThisChartNum, col="red", type = "p")

k=length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
ThisChartNum_LINE_PSA_Data = rbind (ThisChartNum_LINE_PSA_Data, ThisChartNum_LINE_PSA_Data[k,])
k=k+1
ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE] = 10 * 365 # maximum date used here

if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate]) )
{
  ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = 100
}
else
{
  ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA]
}

### points (ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE],ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_PSA], main=ThisChartNum, col="blue", type = "l")

# close the output file
### dev.off()

# examine the PSA values of all the points we kept
NumPoints = length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
# until we have looked at them all or found the before/after gap we were looking for
DONE=FALSE

for (j in 1:NumPoints)
{
  # if the plot starts before the date of the start window
  if (ThisChartNum_LINE_PSA_Data[1,LINE_PSA_Data_DATE] < START)
  {
    # if we find a PSA that is to the right of the start window date
    if (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] > START && !DONE)
    {
      LINE_Start_Bucket[i,Bucket_PreScore] = ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_PSA] # psa before start date
      LINE_Start_Bucket[i,Bucket_PostScore] = ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_PSA] # psa after start date

      DY = 1.0 * (LINE_Start_Bucket[i,Bucket_PostScore] - LINE_Start_Bucket[i,Bucket_PreScore]) # after psa - before psa
      DX = 1.0 * (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] - ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_DATE]) # after time - before
time
      LINE_Start_Bucket[i,Bucket_NowScore] = LINE_Start_Bucket[i,Bucket_PreScore] + DY/DX * as.integer((START - ThisChartNum_LINE_PSA_Data[j-
1,LINE_PSA_Data_DATE]))
      DONE=TRUE
    }
  }
}

# find the values above the minimum
LINE_Start_Bucket_low= rbind(LINE_Start_Bucket[LINE_Start_Bucket[,Bucket_NowScore] < topbuffer , ])
# find the values below the maximum
LINE_Start_Bucket_high= rbind(LINE_Start_Bucket[LINE_Start_Bucket[,Bucket_NowScore] > bottombuffer , ])
# fine the value that are in both of these groups
LINE_Start_Bucket = rbind(LINE_Start_Bucket_low [LINE_Start_Bucket_low[,Bucket_CHART] %in% (LINE_Start_Bucket_high[,Bucket_CHART]),])
}

#####DONE :FIND Charts that match START parameters #####

result_dates = NULL
result_values = NULL
## calculate each End Bucket result

END_Graph = 1.1 * END
if (END_Graph > 3650) END_Graph = 3650

dates = seq(START, END_Graph, DATESTEP)

for (i in dates)
{
  ThisDate = i
  ##
  ##

##### Find Charts Matching Future Parameter #####
##
## if we have at least one patient who matched the start condition
##
## if (0<length(LINE_Start_Bucket[,Bucket_CHART]))
{
  ## find the patient data for all those charts that matched the start parameter (+/- 50%)
ProtoPatData = ProtoPatData [ProtoPatData[,ProtoPatData_CHART] %in% (LINE_Start_Bucket[,Bucket_CHART]),]
# find the PSA data for all those charts that matched the start parameter (+/- 50%)
ProtoPSAData = ProtoPSAData [ProtoPSAData[,ProtoPSAData_CHART] %in% (ProtoPatData[,ProtoPatData_CHART]),]

## makes a more streamlined version of the PSA data with only Chart#/Date/PSA
LINE_PSA_Data = cbind(ProtoPSAData[,ProtoPSAData_CHART], ProtoPSAData[,ProtoPSAData_DATE], ProtoPSAData[,ProtoPSAData_PSA])

```

```

## make a start bucket scratchpad matrix with only Chart#/PSA-left/PSA-right/PSA-now/fit_indicator
LINE_End_Bucket = cbind(ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1],ProtoPatData[,1])
LINE_End_Bucket[,2:4] = -1

# examine each patient that crossed the first criteria, to see if they exceed the second criteria
for (i in 1: length(ProtoPatData[,ProtoPatData_CHART]))
{
  # get the chartnumber for next patient to examine
  ThisChartNum = ProtoPatData[i,ProtoPatData_CHART]
  # get the PSA data for this patient only
  ThisChartNum_LINE_PSA_Data = LINE_PSA_Data[LINE_PSA_Data[,LINE_PSA_Data_CHART]==ThisChartNum,]

  # sort PSA points by date
  ThisChartNum_LINE_PSA_Data = ThisChartNum_LINE_PSA_Data[order(as.numeric(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_DATE])),]

  k=length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])
  ThisChartNum_LINE_PSA_Data = rbind( ThisChartNum_LINE_PSA_Data, ThisChartNum_LINE_PSA_Data[k,])
  k=k+1
  ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_DATE] = 10 * 365 # maximum date used here

  if ( 0 < (ProtoPatData[ ProtoPatData[,ProtoPatData_CHART] == ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_CHART] ,ProtoPatData_INTdate]) )
  {
    ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = 100
  }
  else
  {
    ThisChartNum_LINE_PSA_Data[k,LINE_PSA_Data_PSA] = ThisChartNum_LINE_PSA_Data[k-1,LINE_PSA_Data_PSA]
  }
}

#how many points are we keeping?
NumPoints = length(ThisChartNum_LINE_PSA_Data[,LINE_PSA_Data_CHART])

DONE=FALSE
# until done, look through all the points and estimate the value of the chart at the precise day of interest
for (j in 1:NumPoints)
{
  if (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] > ThisDate && !DONE)
  {
    if (j > 1)
    {
      LINE_End_Bucket[i,Bucket_Prescore] = ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_PSA] # psa before ThisDate
      LINE_End_Bucket[i,Bucket_PostScore] = ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_PSA] # psa after ThisDate

      DY = 1.0 * (LINE_End_Bucket[i,Bucket_PostScore] - LINE_End_Bucket[i,Bucket_Prescore]) # after psa - before psa
      DX = 1.0 * (ThisChartNum_LINE_PSA_Data[j,LINE_PSA_Data_DATE] - ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_DATE]) # after time - before
      time
      LINE_End_Bucket[i,Bucket_NowScore] = LINE_End_Bucket[i,Bucket_Prescore] + DY/DX * as.integer((ThisDate - ThisChartNum_LINE_PSA_Data[j-1,LINE_PSA_Data_DATE]))
    }
    DONE=TRUE
  }
}

# we only want to keep those points that exceeded the end value selected
LINE_End_Bucket_over= rbind(LINE_End_Bucket[LINE_End_Bucket[,Bucket_NowScore] > ENDV , ]) # must rbind to keep 2 dimensions

##### how many exceeded the end value % #####
{
  Denominator = length(LINE_Start_Bucket[,Bucket_CHART])
  Numerator = length(LINE_End_Bucket_over[,Bucket_CHART])

  if (0 == Denominator)
  {
    Result = -1
  }
  else
  {
    Result = (Numerator) / (Denominator * 1.0)
  }
}

result_dates = c(result_dates, as.integer((ThisDate+MonthInDays/2)/MonthInDays))
result_values = c(result_values, 100*Result)
} ## for i in dates .. START,END+DATESTEP,DATESTEP

win.metafile( (paste("chartodds.wmf"))) # prepare plot device

poly_data = data.frame ( cbind( result_values , result_dates ) )

PolyModel = lm (result_values ~ result_dates + I(result_dates^2) ) #2 degree poly **BEST**

par(font.axis=10)
plot(result_dates,result_values, main="Exceed Odds From Previous Data", col = "blue", type = "p", xlab="Months Since RT", ylab="%")
poly.curve <- function(x) predict(PolyModel, data.frame(result_dates=c(x) ), type = "response")

```

```
curve( poly.curve(x), add = TRUE , col="blue")
} ## if end > start
else
{
  par(font.axis=10)
  win.metafile( paste("chartodds.wmf")) # prepare plot device
  plot(100,100, main="ERROR:End Date Not After Start Date", col = "blue", type = "n", xlab="Days", ylab="%")
}
}

# if a plotting device (file) is open, then close it
if (l<dev.cur())
dev.off()
```

APPENDIX E – ‘C++’ “main” code for GUI Control

```

-----
#include <vcl.h>
#pragma hdrstop
#include "Main.h"
#include "About.h"
#include "stdio.h"
-----
#pragma resource "*.dfm"
TMainForm *MainForm;
int TermView = TRUE;
int FreshLoad = TRUE;
-----

__fastcall TMainForm::TMainForm(TComponent *Owner)
: TForm(Owner)
{
}
-----

void __fastcall TMainForm::CreateMDIChild(String Name)
{
    TMDIChild *Child;

    ----- create a new MDI child window -----
    Child = new TMDIChild(Application);
    Child->Caption = Name;
    if (FileExists (Name))
        Child->Memol->Lines->LoadFromFile(Name);
}
-----

void __fastcall TMainForm::FileNewlExecute(TObject *Sender)
{
    CreateMDIChild("NONAME" + IntToStr(MDIChildCount + 1));
}
-----

void __fastcall TMainForm::FileOpenlExecute(TObject *Sender)
{
    AnsiString FileName, Extention = ".wmf";
    int WMF_SubString;
    FILE * FP;

    ForceCurrentDirectory = TRUE;

    if (OpenDialog->Execute())
        CreateMDIChild(OpenDialog->FileName);

    FileName = OpenDialog->FileName;
    WMF_SubString = FileName.AnsiPos(Extention);

    // if a chartfile is selected then display it to the user
    if (0 < WMF_SubString)
    {
        if ((FP = fopen(FileName.c_str(), "r")) == NULL )
            MessageBox(NULL,"File could not be opened", "Error", MB_OK);
        else
        {
            Image1->Picture->LoadFromFile(FileName);
            fclose(FP);
        }
    }
}
-----

void __fastcall TMainForm::HelpAboutlExecute(TObject *Sender)
{
    AboutBox->ShowModal();
}
-----

void __fastcall TMainForm::FileExitlExecute(TObject *Sender)

```

```

{
    Close();
}
//-----

void __fastcall TMainForm::Edit2Change(TObject *Sender)
{
    static float Edit2_Value;
    static AnsiString Edit2_Previous = "0.0";

    // ensure that the now PSA is a valid number
    if (!TryStrToFloat(Edit2->Text, Edit2_Value))
    {
        Edit2->Text = Edit2_Previous;
    }
    else
    {
        Edit2_Previous = Edit2->Text;
    }
}
//-----

void __fastcall TMainForm::Edit3Change(TObject *Sender)
{
    static float Edit3_Value;
    static AnsiString Edit3_Previous = "0.0";

    // ensure that the future PSA is a valid number
    if (!TryStrToFloat(Edit3->Text, Edit3_Value))
    {
        Edit2->Text = Edit3_Previous;
    }
    else
    {
        Edit3_Previous = Edit3->Text;
    }
}
//-----

void __fastcall TMainForm::Button1Click(TObject *Sender)
{
    bool ParameterErrors = FALSE;
    int NowDate = 0, FutureDate = 0;
    float NowPSA = 0, FuturePSA = 0;
    FILE *OutParams, *InParams;
    char OutString[80], InString[80], Edit1_Text[80], Edit4_Text[80];
    int ChartOver = 0, ChartTotal = 0, i;
    float ChartPercent = 0;

    // is a valid risk level selected?
    if (-1 == ComboBox1->ItemIndex)
    {
        ParameterErrors = TRUE;
        MessageBox(NULL, "Error: Risk Level Is Invalid", "Error", MB_OK);
    }

    NowDate = int(DateTimePicker2->Date - DateTimePicker1->Date); // months since RT
    FutureDate = int(DateTimePicker3->Date - DateTimePicker1->Date); // months since RT

    // the future date must be greater than the present date, and both non-negative
    if ( NowDate < 0
        || FutureDate < 0
        || (FutureDate - NowDate) < 0)
    {
        ParameterErrors = TRUE;
        MessageBox(NULL, "Error: Date Parameters Are Invalid", "Error", MB_OK);
    }

    if (NowDate > 10*365)
    {
        NowDate = 10*365 -10; // don't calculate past 10 years
    }

    if (FutureDate > 10*365)
    {
        FutureDate = 10*365 -10; // don't calculate past 10 years
    }

    // is the present date valid?
    if (!TryStrToFloat(Edit2->Text, NowPSA))
    {
        ParameterErrors = TRUE;
        MessageBox(NULL, "Error: Today PSA Is Invalid", "Error", MB_OK);
    }
}

```

```

}
else
{
    sprintf(OutString,"%f",NowPSA);
    Edit2->Text = OutString;
}

// is the future date valid?
if (!TryStrToFloat(Edit3->Text, FuturePSA))
{
    ParameterErrors = TRUE;
    MessageBox(NULL,"Error: Future PSA Is Invalid", "Error", MB_OK);
}
else
{
    sprintf(OutString,"%f",FuturePSA);
    Edit3->Text = OutString;
}

// if there are no input parameter errors then continue to do calculations
if (!ParameterErrors)
{
    // clear the list of patient charts
    ListBox1->Items->Clear();
    FreshLoad = FALSE;
    OutParams = fopen("Rparams.csv","w");

    if (NULL == OutParams)
    {
        ParameterErrors = TRUE;
        MessageBox(NULL,"Error: Cannot Create File For R Subroutine", "Error", MB_OK);
    }
    else
    {
        // put the calculation parameters in a file for R to read
        fprintf(OutParams,"%i,%i,%3.2f,%i,%3.2f,NoErr\n",ComboBox1->ItemIndex,NowDate,NowPSA,FutureDate,FuturePSA);
        fclose(OutParams);
    }

    // calculate the term and future date results based on these parameters
    system("Rscript RfromC_shortview.R");
    system("Rscript RfromC_longview.R");

    // depending on the view chosen, post the graph to the screen
    if (TermView)
    {
        Image1->Picture->LoadFromFile("chartodds.wmf");
    }
    else
    {
        Image1->Picture->LoadFromFile("chartexceeded.wmf");
    }

    InParams = fopen("Cparams.txt","r");
    if (NULL == InParams)
    {
        ParameterErrors = TRUE;
        MessageBox(NULL,"Error: Cannot Open Result File From R Subroutine", "Error", MB_OK);
    }
    else
    {
        // number of charts over threshold is 4th string in result file
        for (i = 0; i <4; i++)
        {
            fscanf(InParams,"%s",InString);
        }
        sscanf(InString, "%i", &ChartOver);

        // number of charts (total) is 8th string in result file
        for (i = 0; i <4; i++)
        {
            fscanf(InParams,"%s",InString);
        }
        sscanf(InString, "%i", &ChartTotal);

        // the list of chart numbers that exceeded the threshold
        while ( EOF != fscanf(InParams,"%s",InString) )
        {
            if ( EOF != fscanf(InParams,"%s",InString) )
            {
                ListBox1->Items->Add(InString);
            }
        }

        fclose(InParams);
    }

    // post the chart counts to the screen

```

```
        sprintf(Edit1_Text, "%i", ChartOver);
        Edit1->Text = Edit1_Text;
        sprintf(Edit4_Text, "%i", ChartTotal);
        Edit4->Text = Edit4_Text;
    }
}
//-----

void __fastcall TMainForm::FileCloseItemClick(TObject *Sender)
{
    /// Image1->Picture->LoadFromFile("blank_chart.wmf");
}
//-----

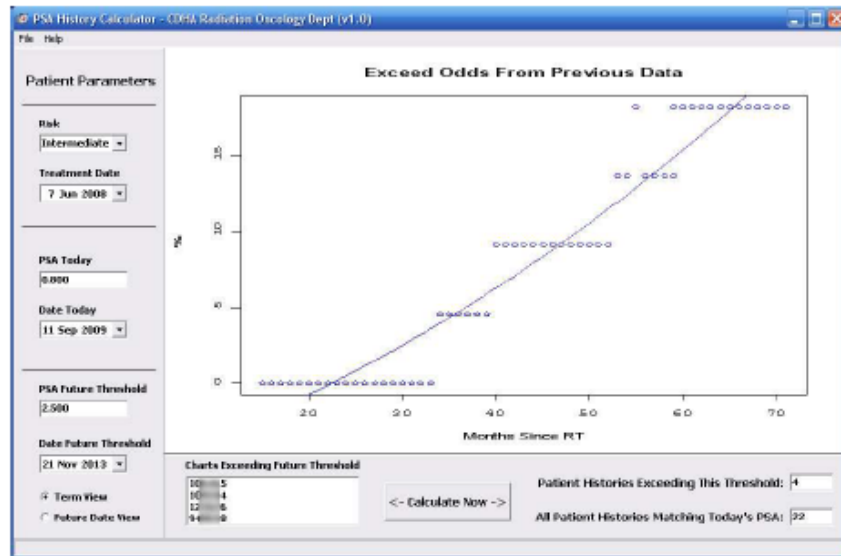
void __fastcall TMainForm::RadioButton1Click(TObject *Sender)
{
    TermView = TRUE;

    // display the term view
    if (!FreshLoad)
    {
        Image1->Picture->LoadFromFile("chartodds.wmf");
    }
}
//-----

void __fastcall TMainForm::RadioButton2Click(TObject *Sender)
{
    TermView = FALSE;

    // display the future date view
    if (!FreshLoad)
    {
        Image1->Picture->LoadFromFile("chartexceeded.wmf");
    }
}
//-----
```

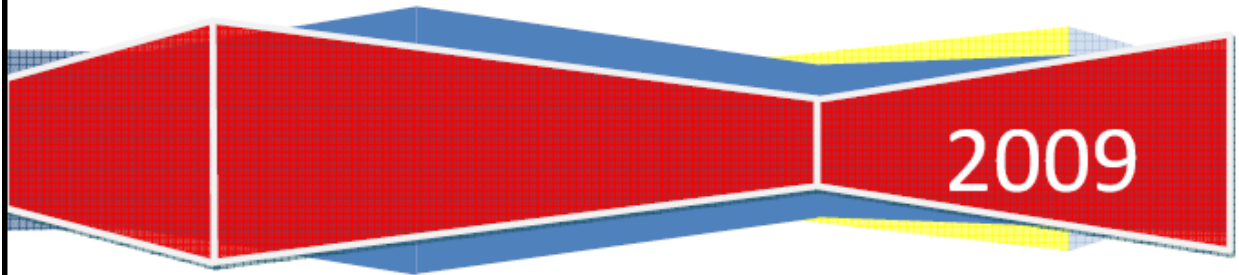

APPENDIX F – PSA History Calculator User Guide



PSA HISTORY CALCULATOR

CDHA Radiation Oncology Dept

Tom Harding



PSA History Calculator



Contents

Introduction	3
Background	3
Installation	4
Usage	5
Development Tools	8
Data Preparation	10
References	16

PSA History Calculator



Introduction

The PSA levels of patients with prostate cancer who have been treated by radiation therapy are monitored periodically before and after treatment. A rise in these levels can indicate a relapse long in advance of other symptoms. For this reason it is valuable to calculate the probability of a rise in PSA levels when considering scheduling decisions for follow-up appointments. This PSA Calculator application is developed specifically for this purpose. (In all diagrams in this guide, the 3rd, 4th, and 5th digits of any patient chart number are purposely blurred out.)

Background

An initial database for capturing the prognostic factors for patients with prostate cancer was first created by Dr Rutledge in the late 90's. Since that time, it has also been used by Dr. Sun and Dr. Wilke and now contains history information for 100's of patients. This data has been used as the case history information for calculations by this PSA History Calculator (PHC).

This application requires 7 information attributes in two files as source data. Those attributes are patient chart numbers, treating oncologist, risk level, radiation therapy treatment date, date of ADT intervention (if applicable), PSA test and PSA test results. Although more attributes are available, further subdividing the data rapidly reduced the statistical weight of set calculations. Therefore, in this early version of the application, patient set data divisions are kept simple (PSA and Patient Risk Level) in order to retain a significant case basis for each calculation. If more data becomes available in the future, this approach could be changed to add more patient criteria per calculation.

PSA History Calculator



Installation

Installing the PHC on a PC station requires two primary steps:

Install the 'R' statistical software:

- On the original delivered CD, there is a subfolder named 'R'. In this subfolder is the installation package for 'R' called 'R-2.9.1-win32'. This package must be installed on any PC where PHC will be installed. Double click on the 'R-2.9.1-win32' icon and follow the prompts.
- After installing 'R' the path to the folder containing 'Rscript.exe' must be added to the PC's path statement. For windows 2000 or XP this is done by right clicking on 'My Computer' and then selecting properties, and then clicking on the tab called 'Advanced', and then the button called 'Environmental Variables'. On the 'Environmental Variables' page, select the path variable and edit it to include the name of the folder where you installed the 'R' bin (binary) files.

Install the 'PHC' executable software:

- Under the 'Program Files' folder create a subfolder called 'PHC'. Copy all files from the original delivered CD disc 'PHC' subfolder into this new folder.
- Create a desktop icon shortcut to the executable file in the new 'PHC' folder named 'PSA Hist Calc'.

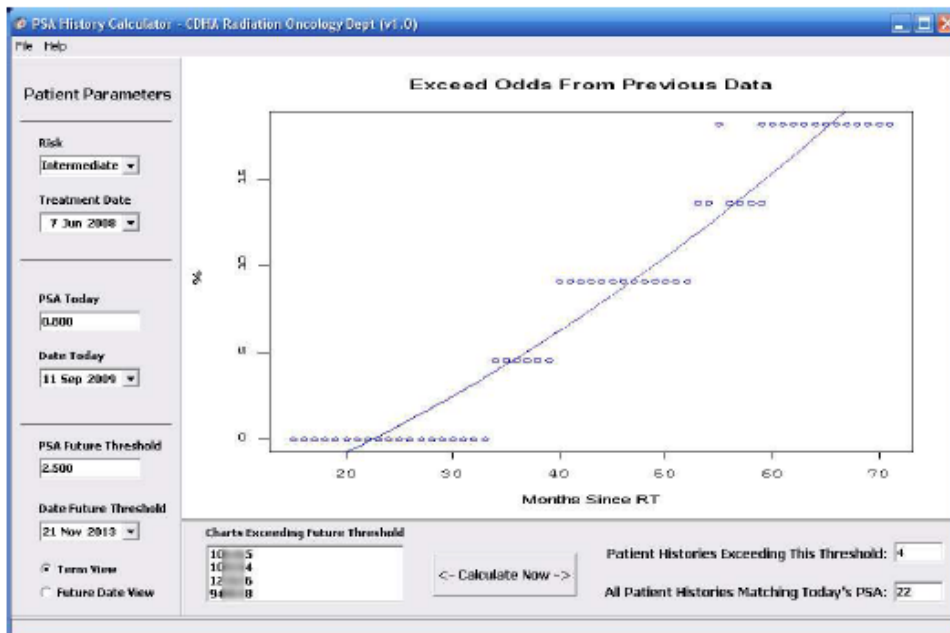


PSA History Calculator

Usage

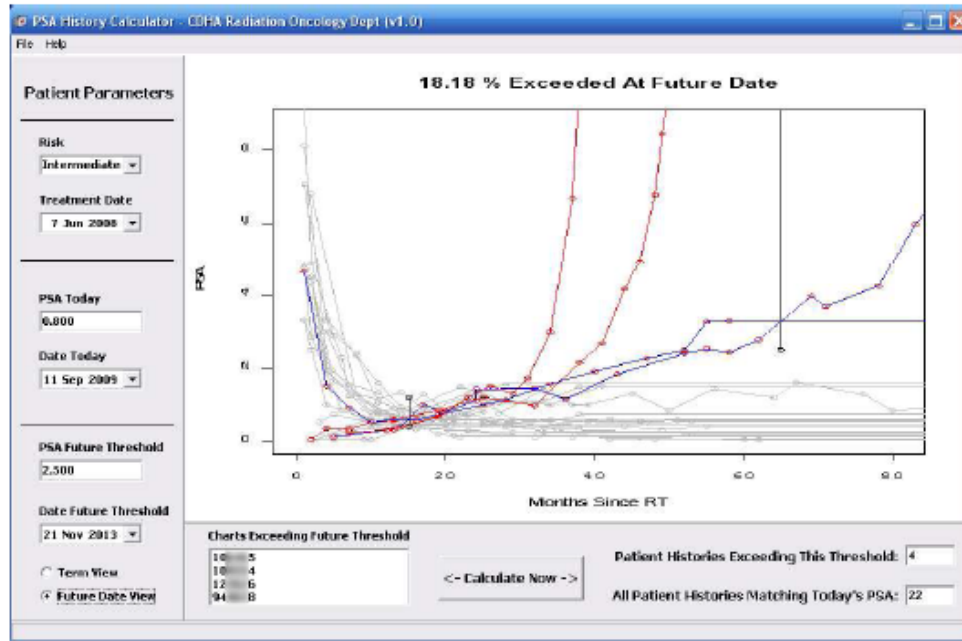
This application requires the following input:

- a) Patient Risk Level
- b) The date of this patient's RT treatment. This field can be selected by using the pull-down and scrolling to the date of interest, or each field (day, month, year) can be typed in manually. For months, use the number of the month, so for April you would press "4" to see April.
- c) The PSA of the patient's most recent test score (today).
- d) The Date of the patient's most recent test score (today).
- e) The PSA of comparative interest at some future date.
- f) The Date in the future for comparative results.
- g) Term View (% of past histories which exceeded the comparative PSA) over time, or Future Date View (how many case histories were below or above the comparative value on the future date chosen).



This term view result shows a result of comparing a patient profile with historic data. In this case the number of cases which exceeded the Future Threshold of 2.5 gradually rises from 0 to about 18% at the future date selected.

PSA History Calculator



This Future Date view shows that considering the initial window of interest at RT date plus 15 months for patients with a PSA of 0.8 (+/- 50%), 18.18% will go on to exceed the Future Threshold of 2.5 before RT date plus 65 months. Two of those (red lines) were due to treatment failure and required ADT. The chart number of the 4 cases which exceeded the Future Threshold are shown in the Chart Exceeding window near the bottom center of the display. Any particular chart plot of interest can be viewed by using File>Open>Chartname as shown below:

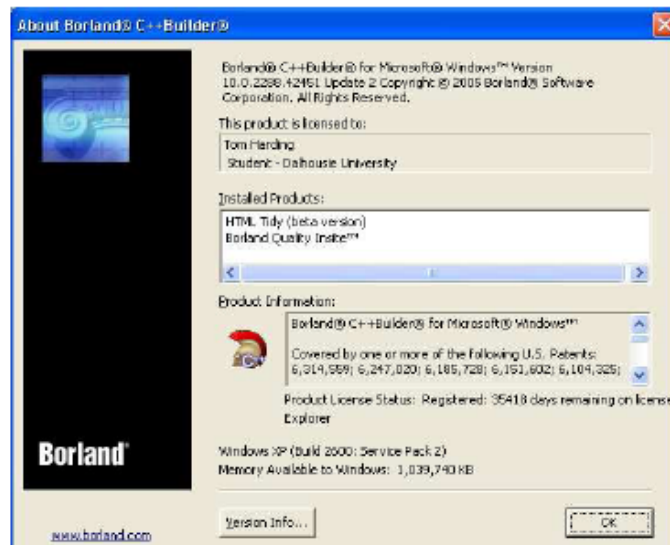
PSA History Calculator



Development Tools

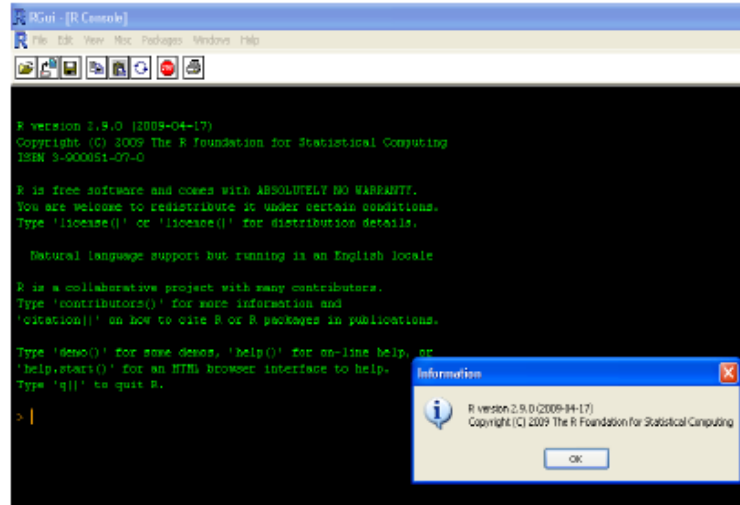
This application was developed using three primary tools. Those are Microsoft Excel, Borland C++, and R (www.r-project.org). The rationale and specific versions are as follows:

- a) Microsoft Excel Version 2007 was used for filtering out data with corrupt or missing fields, and for converting dates from Gregorian calendar format to Julian. Julian dates make for a more consistent measure of time for data spanning many months or years.
- b) Borland C++ was used for development of the graphic user interface (GUI) because 'R' does not support user input or output in a windows environment, and 'R' can be difficult to operate for anyone who has not spent a considerable amount of time learning its use.



- c) 'R' was used for the statistical and data plotting engine in this application because it is more capable than most commercial statistical packages and it is free for use. This gives the final application speedy calculations on large volumes of tabular data without the need for licensing fees often associated with such software.

PSA History Calculator



PSA History Calculator



Data Preparation

These steps are included for the contingency that someone would like to add or update the data which this tool is using as the basis of its calculations. This section of instructions may be ignored until that time. (In all of these diagrams the 3rd, 4th, and 5th digits of any patient chart number are purposely blurred out.)

- a) First a patient data file must be tailored to include corresponding chart numbers, risk level, treating radiation oncologist, start of RT date, and ADT intervention date (if applicable). All data in columns A, B, and D must be valid and non-blank. All data in column E must be either a valid date or blank. Data in column C is free format and can be any value or none.

	A	B	C	D	E	F	G
1	chart	risk_strata	treating_rad_oncol	start_RT	interven_ADT_Date		
2	12-11-4	intermediate	Derek R Wilke	28-May-04			
3	12-12-7	intermediate	Derek R Wilke	3-May-04			
4	12-12-3	intermediate	Derek R Wilke	26-Jul-04			
5	12-12-9	intermediate	Derek R Wilke	7-May-04			
6	12-12-1	intermediate	Derek R Wilke	7-Jul-04			
7	12-12-9	high	DW	30-Jun-04			
8	12-12-1	intermediate	Derek R Wilke	1-Jun-05			
9	12-12-9	low	Derek R Wilke	27-Apr-04			
10	12-12-8	high	Derek R Wilke	21-Sep-04			
11	12-12-0	intermediate	Derek R Wilke	7-Jun-04			
12	12-12-7	high	Derek R Wilke	16-Sep-04			
13	12-12-6	intermediate	Derek R Wilke	3-Mar-05			
14	12-12-5	intermediate	Derek R Wilke	29-Nov-04			
15	12-12-1	intermediate	DW	22-Sep-04			
16	12-12-7	intermediate	Derek R Wilke	23-Sep-04			
17	12-12-8	intermediate	Derek R Wilke	25-Jan-05			
18	12-12-0	low	Derek R Wilke	18-May-05			
19	12-12-3	high	DW	21-Apr-05			
20	12-12-3	intermediate	Derek R Wilke	30-Dec-04			
21	12-12-2	high	Derek R Wilke	24-Feb-05	18-Apr-06		
22	12-12-8	intermediate	Derek R Wilke	25-Apr-05			
23	12-12-6	high	DW	4-Jul-05			

PSA History Calculator



- b) Next 4 columns must be inserted right of the RT treatment date and another 4 columns right of the ADT intervention dates.

chart	risk_strata	treating_rad_onc_data	start_RT	day_RT	month_RT	year_RT	julian_RT	Interven_ADT	day_ADT	month_ADT	year_ADT	julian_int_ADT
1	11	4	Derek R Wilko	29	May	04						
2	12	7	Derek R Wilko	9	May	04						
3	12	8	Derek R Wilko	26	Jul	04						
4	12	8	Derek R Wilko	7	May	05						
5	12	8	Derek R Wilko	7	Jul	04						
6	12	10	DW	30	Jul	04						
7	12	10	DW	1	Jun	04						
8	12	10	Derek R Wilko	27	Apr	04						
9	12	10	Derek R Wilko	21	Sep	04						
10	12	10	Derek R Wilko	7	Jun	04						
11	12	10	Derek R Wilko	16	Sep	04						
12	12	10	Derek R Wilko	3	Mar	05						
13	12	10	Derek R Wilko	29	Nov	04						
14	12	10	DW	22	Sep	04						
15	12	10	Derek R Wilko	29	Sep	04						
16	12	10	Derek R Wilko	25	Jan	05						
17	12	10	Derek R Wilko	16	May	05						
18	12	10	DW	21	Apr	05						
19	12	10	Derek R Wilko	30	Dec	04						
20	12	10	Derek R Wilko	24	Feb	05						
21	12	10	Derek R Wilko	23	Apr	05						
22	12	10	DW	4	Jul	05						
23	12	10	Derek R Wilko	27	Apr	05						
24	12	10	Derek R Wilko	12	Apr	05						
25	12	10	Derek R Wilko	6	Jun	05						
26	12	10	Derek R Wilko	20	May	05						

PSA History Calculator



- c) Next the formula to extract the day of the month must be entered in cell E2. (“=DAY(D2)”)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N												
	chart	risk_strata	treating_rad	on_date	start	RT	day	RT	month	RT	year	RT	julian	RT	interven	ADT	day	ADT	month	ADT	year	ADT	julian	int	ADT	
2	120004	intermediate	Derek R Wilke	26-May-04																						
3	120007	intermediate	Derek R Wilke	3-May-04																						
4	120005	intermediate	Derek R Wilke	26-Jul-04																						
5	120008	intermediate	Derek R Wilke	7-May-04																						
6	120001	intermediate	Derek R Wilke	7-Jul-04																						
7	120010	high	DW	30-Jun-04																						
8	120001	intermediate	Derek R Wilke	1-Jun-04																						
9	120009	low	Derek R Wilke	27-Apr-04																						
10	120006	high	Derek R Wilke	21-Sep-04																						
11	120010	intermediate	Derek R Wilke	7-Jun-04																						
12	120007	high	Derek R Wilke	16-Sep-04																						
13	120000	intermediate	Derek R Wilke	1-Mar-05																						
14	120005	intermediate	Derek R Wilke	29-Nov-04																						
15	120004	intermediate	DW	22-Sep-04																						
16	120007	intermediate	Derek R Wilke	29-Sep-04																						
17	120008	intermediate	Derek R Wilke	25-Jan-05																						
18	120010	low	Derek R Wilke	16-May-05																						
19	120010	high	DW	22-Apr-05																						
20	120005	intermediate	Derek R Wilke	30-Dec-04																						
21	120002	high	Derek R Wilke	24-Feb-05																						
22	120006	intermediate	Derek R Wilke	25-Apr-05																						
23	120010	high	DW	4-Jul-05																						
24	120004	high	Derek R Wilke	27-Apr-05																						
25	120006	intermediate	Derek R Wilke	22-Apr-05																						
26	120006	intermediate	Derek R Wilke	6-Jun-05																						
27	120006	intermediate	Derek R Wilke	30-May-05																						

- d) Next the formula to extract month of the year must be entered in cell F2. (“=MONTH(D2)”)
- e) Next the formula to extract year of the date must be entered in cell G2. (“=YEAR(D2)”)
- f) Then enter the formula to calculate the Julian date in cell H2.
 (“= (367 * G2) - INT(7*(G2 + INT((F2 + 9)/12))/4) + INT(275*F2/9) + E2 + 1721013”)
- g) Select cells E2 through G2 (there will be a very small black box in the bottom right corner of the select box you create when you select this group of cells, drag it) and drag these formulas all the way down columns E, F, G, and H till they have been copied all the way to the bottom of the entire spreadsheet.

PSA History Calculator



1	chart	rlk_status	treatment	start_date	start_RT	day_RT	month_RT	year_RT	Julian_RT	interven_A01day_A01	month_A01	year_A01	Julian_int_A01
2	12114	intermediate	Derek R Wilkie	18-May-04	28	5	2004	245113					
3	12117	intermediate	Derek R Wilkie	3-May-04	3	5	2004	245113					
4	12112	intermediate	Derek R Wilkie	20-Jul-04	20	7	2004	245206					
5	12110	intermediate	Derek R Wilkie	3-May-04	7	5	2004	245212					
6	12111	intermediate	Derek R Wilkie	7-Jul-04	7	7	2004	245315					
7	12110	high	DW	16-Jan-04	16	0	2004	245318					
8	12111	intermediate	Derek R Wilkie	1-Jul-04	1	0	2004	245317					
9	12110	low	Derek R Wilkie	27-Apr-04	27	4	2004	245322					
10	12110	high	Derek R Wilkie	21-Sep-04	21	9	2004	245326					
11	12110	intermediate	Derek R Wilkie	3-Jul-04	7	0	2004	245326					
12	12117	high	Derek R Wilkie	16-Sep-04	16	9	2004	245326					
13	12110	intermediate	Derek R Wilkie	3-Mar-05	3	3	2005	245342					
14	12110	intermediate	Derek R Wilkie	20-Nov-04	20	11	2004	245328					
15	12111	intermediate	DW	22-Sep-04	22	9	2004	245327					
16	12117	intermediate	Derek R Wilkie	23-Sep-04	23	9	2004	245327					
17	12110	intermediate	Derek R Wilkie	25-Jan-05	25	1	2005	245355					
18	12110	low	Derek R Wilkie	18-May-05	18	5	2005	245356					
19	12110	high	DW	21-Apr-05	21	4	2005	245481					
20	12110	intermediate	Derek R Wilkie	30-Dec-04	30	12	2004	245369					
21	12112	high	Derek R Wilkie	26-Feb-05	26	2	2005	245425	18-Apr-05				
22	12110	intermediate	Derek R Wilkie	25-Apr-05	25	4	2005	245485					
23	12110	high	DW	4-Jul-05	4	7	2005	245255					
24	12114	high	Derek R Wilkie	27-Apr-05	27	4	2005	245487					
25	12110	intermediate	Derek R Wilkie	12-Apr-05	12	4	2005	245472					
26	12110	intermediate	Derek R Wilkie	8-Jul-05	8	0	2005	245527					
27	12110	intermediate	Derek R Wilkie	18-May-05	18	5	2005	245520					

h) Again select cells E2 through H2 and copy them to cells J2 through M2. Then select J2 through M2 and drag the column formulas to the bottom of the entire spreadsheet again.

1	chart	rlk_status	treatment	start_date	start_RT	day_RT	month_RT	year_RT	Julian_RT	interven_A01day_A01	month_A01	year_A01	Julian_int_A01
2	12114	intermediate	Derek R Wilkie	18-May-04	28	5	2004	245113		0	1	1900	2455018
3	12117	intermediate	Derek R Wilkie	3-May-04	3	5	2004	245113		0	1	1900	2455018
4	12112	intermediate	Derek R Wilkie	20-Jul-04	20	7	2004	245206		0	1	1900	2455018
5	12110	intermediate	Derek R Wilkie	3-May-04	7	5	2004	245212		0	1	1900	2455018
6	12111	intermediate	Derek R Wilkie	7-Jul-04	7	7	2004	245315		0	1	1900	2455018
7	12110	high	DW	16-Jan-04	16	0	2004	245318		0	1	1900	2455018
8	12111	intermediate	Derek R Wilkie	1-Jul-04	1	0	2004	245317		0	1	1900	2455018
9	12110	low	Derek R Wilkie	27-Apr-04	27	4	2004	245322		0	1	1900	2455018
10	12110	high	Derek R Wilkie	21-Sep-04	21	9	2004	245326		0	1	1900	2455018
11	12110	intermediate	Derek R Wilkie	3-Jul-04	7	0	2004	245326		0	1	1900	2455018
12	12117	high	Derek R Wilkie	16-Sep-04	16	9	2004	245326		0	1	1900	2455018
13	12110	intermediate	Derek R Wilkie	3-Mar-05	3	3	2005	245342		0	1	1900	2455018
14	12110	intermediate	Derek R Wilkie	20-Nov-04	20	11	2004	245328		0	1	1900	2455018
15	12111	intermediate	DW	22-Sep-04	22	9	2004	245327		0	1	1900	2455018
16	12117	intermediate	Derek R Wilkie	23-Sep-04	23	9	2004	245327		0	1	1900	2455018
17	12110	intermediate	Derek R Wilkie	25-Jan-05	25	1	2005	245355		0	1	1900	2455018
18	12110	low	Derek R Wilkie	18-May-05	18	5	2005	245356		0	1	1900	2455018
19	12110	high	DW	21-Apr-05	21	4	2005	245481		0	1	1900	2455018
20	12110	intermediate	Derek R Wilkie	30-Dec-04	30	12	2004	245369		0	1	1900	2455018
21	12112	high	Derek R Wilkie	26-Feb-05	26	2	2005	245425	18-Apr-05	18	0	2006	2455843
22	12110	intermediate	Derek R Wilkie	25-Apr-05	25	4	2005	245485		0	1	1900	2455018
23	12110	high	DW	4-Jul-05	4	7	2005	245255		0	1	1900	2455018
24	12114	high	Derek R Wilkie	27-Apr-05	27	4	2005	245487		0	1	1900	2455018
25	12110	intermediate	Derek R Wilkie	12-Apr-05	12	4	2005	245472		0	1	1900	2455018
26	12110	intermediate	Derek R Wilkie	8-Jul-05	8	0	2005	245527		0	1	1900	2455018
27	12110	intermediate	Derek R Wilkie	18-May-05	18	5	2005	245520		0	1	1900	2455018



PSA History Calculator

- i) Save the patient data Excel spreadsheet. Save it again as "other formats">"CSV(comma delimited)"
- j) The patient data file is ready for preprocessing. Exit Excel. (Important, you must **exit**.)
- k) Open Excel again and then open the patient data file excel spreadsheet (do not open the CSV formatted copy). Again select cells E2 through H2, and press control-C (IE.copy).
- l) Open the raw PSA data file. It should look similar to this:

	A	B	C	D	E	F	G
1	chart	date	PSA				
2	10 6	3/9/2009 0:00	14.90				
3	10 2	1/19/2000 0:00	9.00				
4	10 2	1/17/2001 0:00	0.81				
5	10 2	7/13/2000 0:00	0.34				
6	10 2	9/7/2001 0:00	0.86				
7	10 2	2/20/2002 0:00	1.20				
8	10 2	7/3/2002 0:00	0.85				
9	10 2	11/14/2002 0:00	1.20				
10	10 2	3/4/2003 0:00	1.00				
11	10 2	8/8/2003 0:00	1.00				
12	10 2	12/11/2003 0:00	0.97				
13	10 2	4/8/2004 0:00	0.91				
14	10 2	8/31/2004 0:00	0.84				
15	10 2	12/10/2004 0:00	1.10				
16	10 2	4/7/2005 0:00	1.16				
17	10 2	9/13/2005 0:00	1.25				
18	10 2	1/11/2006 0:00	1.53				
19	10 2	5/9/2006 0:00	1.50				
20	10 9	6/7/2000 0:00	3.50				
21	10 9	8/2/2000 0:00	1.00				
22	10 9	10/31/2000 0:00	0.43				
23	10 4	2/3/2000 0:00	0.04				
24	10 2	4/22/2008 0:00	0.17				
25	10 2	4/12/2006 0:00	0.30				
26	10 2	4/23/2007 0:00	0.27				

- m) Insert 4 columns to the right of the date column. Select cells C2 through F2 in the PSA file and press control-V (IE.paste). You can also add titles for columns C through F. Select cells C2 through F2 and again drag the formulas all the way to the bottom of the spreadsheet. You should now have a file that looks like this:

PSA History Calculator



	A	B	C	D	E	F	G	H	I	J	K	L	M
	date	date	day	month	year	value	PSA						
2	10/1/02	3/9/2008 0:00	9	3	2008	2454822	14.90						
3	10/1/02	1/19/2008 0:00	19	1	2008	2451962	9.90						
4	10/1/02	1/17/2001 0:00	17	1	2001	2451826	0.81						
5	10/1/02	3/13/2008 0:00	13	3	2008	2451738	0.34						
6	10/1/02	8/7/2001 0:00	7	8	2001	2452169	0.86						
7	10/1/02	2/20/2002 0:00	20	2	2002	2452325	1.20						
8	10/1/02	7/3/2002 0:00	3	7	2002	2452468	0.85						
9	10/1/02	11/14/2002 0:00	14	11	2002	2452592	1.20						
10	10/1/02	3/4/2003 0:00	4	3	2003	2452700	1.80						
11	10/1/02	8/8/2003 0:00	8	8	2003	2452859	1.89						
12	10/1/02	12/11/2003 0:00	11	12	2003	2452884	0.87						
13	10/1/02	4/8/2004 0:00	8	4	2004	2453703	0.81						
14	10/1/02	8/31/2004 0:00	31	8	2004	2453348	0.84						
15	10/1/02	12/19/2004 0:00	19	12	2004	2453349	1.10						
16	10/1/02	4/7/2005 0:00	7	4	2005	2453467	1.95						
17	10/1/02	9/13/2005 0:00	13	9	2005	2453626	1.25						
18	10/1/02	1/11/2006 0:00	11	1	2006	2453746	1.53						
19	10/1/02	5/9/2006 0:00	9	5	2006	2453884	1.50						
20	10/1/02	8/7/2006 0:00	7	8	2006	2451700	3.50						
21	10/1/02	8/2/2008 0:00	2	8	2008	2451758	1.89						
22	10/1/02	10/31/2008 0:00	31	10	2008	2451848	0.43						
23	10/1/04	2/3/2008 0:00	3	2	2008	2451577	0.84						
24	10/1/02	4/22/2008 0:00	22	4	2008	2454578	0.17						
25	10/1/02	8/19/2008 0:00	19	8	2008	2453837	0.30						
26	10/1/02	4/23/2007 0:00	23	4	2007	2454213	0.27						
27	10/1/02	12/3/2007 0:00	3	12	2007	2454437	0.22						
28	10/1/02	4/4/2008 0:00	4	4	2008	2454825	0.18						
29	10/1/06	3/26/2008 0:00	26	3	2008	2454951	0.85						
30	10/1/06	11/23/2005 0:00	23	11	2005	2453897	30.40						

- n) Save the PSA data Excel spreadsheet. Save it again as "other formats">"CSV(comma delimited)"
- o) Exit Excel. Aren't you glad you love MS Excel ? ☺
- p) At this point you will have a patient CSV file, and a PSA CSV file. Make sure the patient one is named "Julian Patient_Data_" and that the PSA one is named "Julian PSA_values_". If they give an error during processing, you can look inside the script named "R Prep Fail Data" (using notepad) and check the file names it is expecting at about lines 35 and 36. The file names and the names inside the script must match.
- q) Run the script named "R Prep Fail Data". This will create 2 new files called "Prepped PatRawData" and "Prepped OurPats_PSA_Data". Copy those files into the PSA Calculators installation directory. The next time the PSA Calculator is called, it will use the new data in these two files. (If you may wonder, why does the application need these files arranged in this way, the reason is that the initial processing of these two files is very time consuming. You may notice that the "R Prep Fail Data" may take 10 or 15 minutes (or longer) to finish. When we want to use the PSA Calculator, we want it to give results quickly, so these two files contain the preprocessed data that can be quickly searched by the PSA Calculator. Otherwise it would take 10 or 15 minutes (or longer) for every calculation.)

PSA History Calculator



References

Primary problem domain background is based on a description by Dr Rutledge and Dr Wilke.

Borland Developer Software homepage is <http://www.borland.com/>

R Open Source Statistical Software is sourced from <http://cran.r-project.org/>

APPENDIX G – Further Data Request and Proposed Direction

Prostate Cancer PSA History Calculator Progression

In Canada each year approximately 18000 men are diagnosed with prostate cancer. However, the 10 year survival rates are in excess of 90% and in those individuals in whom the cancer recurs, most will die of other causes. The prostate specific antigen (PSA) is a tumor marker in a blood test that often signals recurrence of disease but still may be observed years before any intervention is required. Currently, the optimal follow up schedule in this population is unknown. A tool that predicts the likelihood of recurrence and the necessary frequency of PSA monitoring is needed.

Such a tool would be required to forecast when the PSA level and its growth are clinically significant in those patients with prostate cancer treated with radiotherapy (RT); with or without neo-adjuvant androgen deprivation therapy (NAAD). This would be particularly advantageous for patients who have a low probability of recurrence requiring intervention. It would reduce unnecessary follow up appointments and would reduce costs to our taxed health care system.

We have built a prototype software program to map PSA scores over time using our own patient data which has been collected over a period of 10 years. As mentioned above, we are including both patients with and without neo-adjuvant androgen deprivation therapy (NAAD). In our patient dataset most high risk patients are also on long term androgen deprivation therapy (LADT) from the time of initial RT and are therefore not later switched to LADT as may happen in low or intermediate risk cases. We have specifically excluded those patients who had previous prostatectomy.

The data variables in this initial implementation are limited to patient chart number, patient risk level, RT start date, date of PSA test, PSA score and date of post RT LADT. The last variable is the date on which a low or intermediate risk patient with a rapidly rising PSA, indicating post RT PSA failure, was started on LADT to attempt to suppress further PSA growth. Patient data for low or intermediate risk cases where LADT is started for this reason are not plotted past this date as their PSA progression is no longer reflective of a post RT patient case situation only. We do assume that had they not been started on LADT on this date, their PSA would have continued to rise rapidly thereafter.

Our initial raw data set included more than 1000 patients, however after filtering the data in accordance with the criteria described above, we have about 300 patients which fit our inclusion criteria and can be used for calculating PSA behavior. The calculator renders a result based upon the time since RT treatment, the predicted time until future follow-up appointment, patient risk level, and an estimated future PSA expectation. In order for the calculator to have the statistical power to provide a reliable estimate we need about thirty patients per case grouping. This has been especially problematic for low and high risk patients as each case group selected

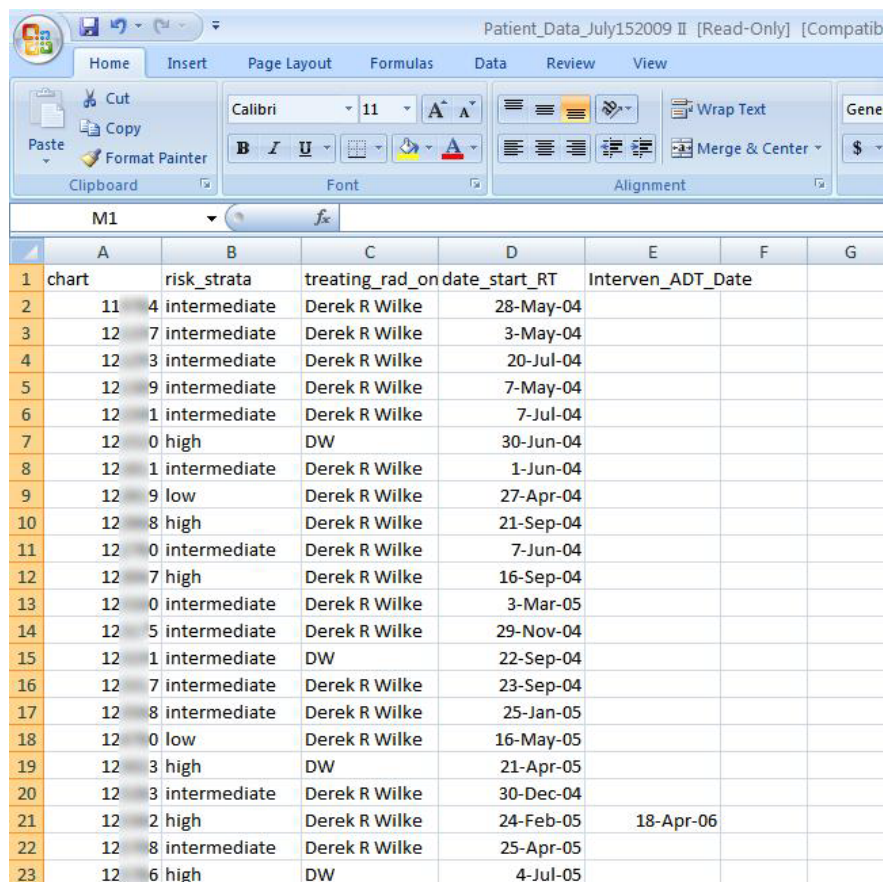
will often yield only about 5 previous histories which are near matches. This can be solved however by increasing the quantity of base historical data that the tool uses by adding more patient data from other establishments that follow similar procedures to our own. We are currently in the process of seeking out such data.

Data from other sources would ideally (but not necessarily) begin as a set of two files: A patient data file and a PSA data file. The records in the files do not need to be sorted. The minimum information content required to support our existing model is as follows (If more information is given than is needed, we can filter the portions out that we wish to work with as we go along.)

The patient data file (first file) contains these attributes :

- ◇the patient chart number (or any uniquely identifying number),
- ◇the patient's risk level,
- ◇the treating RadOnc (this field can be blank),
- ◇the date that the patient started RT,
- ◇the date of ADT intervention if applicable.

Example:



	A	B	C	D	E	F	G
1	chart	risk_strata	treating_rad_on	date_start_RT	Interven_ADT_Date		
2	11	4 intermediate	Derek R Wilke	28-May-04			
3	12	7 intermediate	Derek R Wilke	3-May-04			
4	12	3 intermediate	Derek R Wilke	20-Jul-04			
5	12	9 intermediate	Derek R Wilke	7-May-04			
6	12	1 intermediate	Derek R Wilke	7-Jul-04			
7	12	0 high	DW	30-Jun-04			
8	12	1 intermediate	Derek R Wilke	1-Jun-04			
9	12	9 low	Derek R Wilke	27-Apr-04			
10	12	8 high	Derek R Wilke	21-Sep-04			
11	12	0 intermediate	Derek R Wilke	7-Jun-04			
12	12	7 high	Derek R Wilke	16-Sep-04			
13	12	0 intermediate	Derek R Wilke	3-Mar-05			
14	12	5 intermediate	Derek R Wilke	29-Nov-04			
15	12	1 intermediate	DW	22-Sep-04			
16	12	7 intermediate	Derek R Wilke	23-Sep-04			
17	12	8 intermediate	Derek R Wilke	25-Jan-05			
18	12	0 low	Derek R Wilke	16-May-05			
19	12	3 high	DW	21-Apr-05			
20	12	3 intermediate	Derek R Wilke	30-Dec-04			
21	12	2 high	Derek R Wilke	24-Feb-05	18-Apr-06		
22	12	8 intermediate	Derek R Wilke	25-Apr-05			
23	12	6 high	DW	4-Jul-05			

The PSA data file (second file) contains PSA test results:

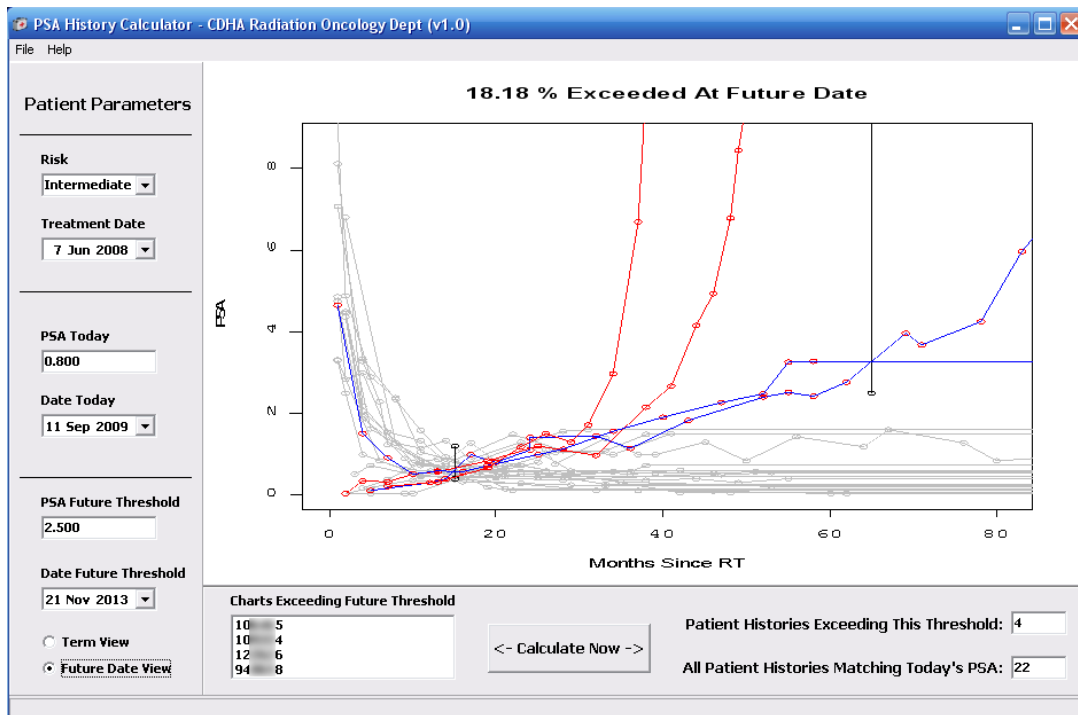
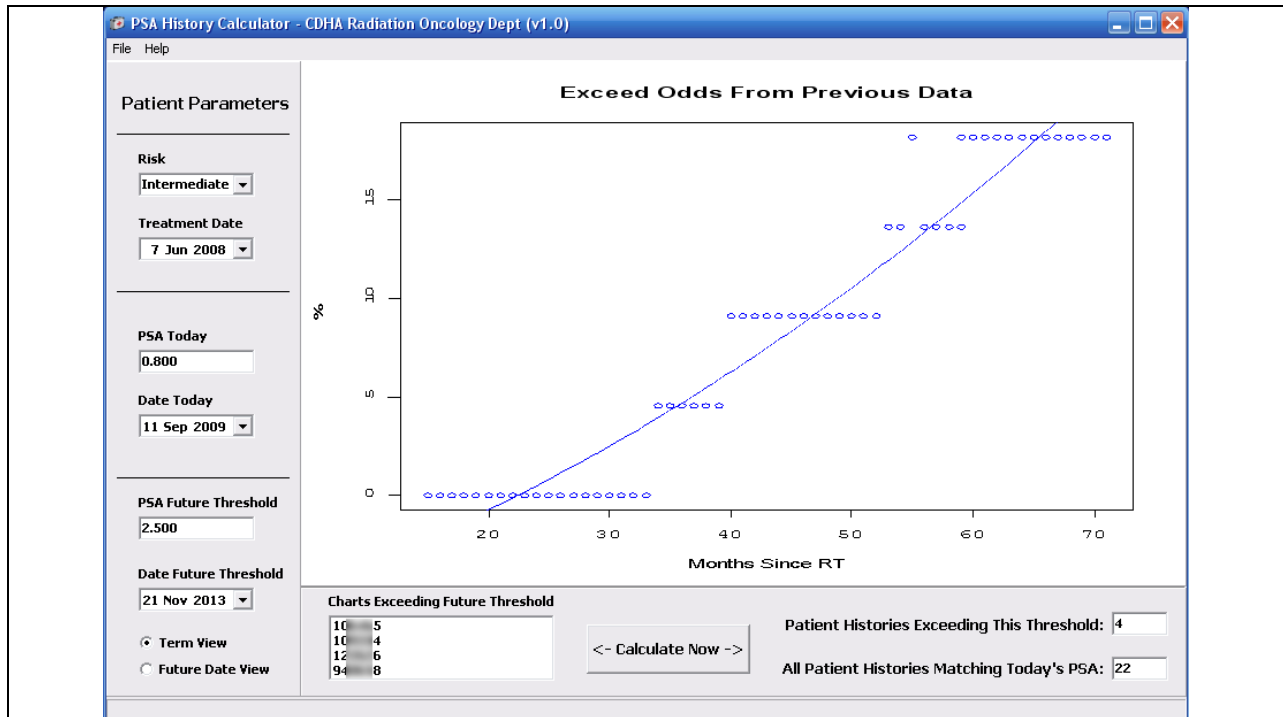
- ◊ the patient chart number (must correspond to patient ID number in the patient data file)
- ◊ the PSA test date
- ◊ the PSA test score

Example :

	A	B	C	D	E	F	G
1	chart	date	PSA				
2	10 5	3/9/2009 0:00	14.90				
3	10 2	1/19/2000 0:00	9.00				
4	10 2	1/17/2001 0:00	0.81				
5	10 2	7/13/2000 0:00	0.34				
6	10 2	9/7/2001 0:00	0.86				
7	10 2	2/20/2002 0:00	1.20				
8	10 2	7/3/2002 0:00	0.85				
9	10 2	11/14/2002 0:00	1.20				
10	10 2	3/4/2003 0:00	1.00				
11	10 2	8/8/2003 0:00	1.00				
12	10 2	12/11/2003 0:00	0.97				
13	10 2	4/8/2004 0:00	0.91				
14	10 2	8/31/2004 0:00	0.84				
15	10 2	12/10/2004 0:00	1.10				
16	10 2	4/7/2005 0:00	1.16				
17	10 2	9/13/2005 0:00	1.25				
18	10 2	1/11/2006 0:00	1.53				
19	10 2	5/9/2006 0:00	1.50				
20	10 9	6/7/2000 0:00	3.50				
21	10 9	8/2/2000 0:00	1.00				
22	10 9	10/31/2000 0:00	0.43				
23	10 4	2/3/2000 0:00	0.04				
24	10 2	4/22/2008 0:00	0.17				
25	10 2	4/12/2006 0:00	0.30				
26	10 2	4/23/2007 0:00	0.27				

... given these 8 attributes as a source data set (or 7 attributes if treating RadOnc name is not available), we can remap the data from other organizations and compare their patient results with our own.

The PSA History Calculator screenshots:



Planned future improvements with increased source data:

By increasing the size of the source patient data in quantity and scope of attributes, we will have

the opportunity to explore other possible ways of drawing results from the data. For example, with enough data we could replace the simple patient risk level criteria with something more specific (example: TMN, Consult PSA, Gleason scores), or completely separate the adjuvant ADT patient set from the nonadjuvant. Each level of refinement can yield results more specifically attuned to the particulars of each patient. An increased dataset will also give us the option to apply data mining principles to the dataset to identify subtle results which are not immediately discernable, and to prototype the use of advanced data classifiers such as C4.5/C5.0 (ID3) decision trees, or bayesian network belief models to predict some outcomes. A larger base model dataset would also facilitate calculation of result confidence intervals and point to the identification and filtering of outlier patient plots that may be skewing results in the current implementation.

We should also be concerned with gathering more source evidence to examine the effects of the calculation assumptions that we have made thus far. We should test these assumptions to see if they create reliable results or if they need to be adjusted for better results. With the need to divide data into train and test sets, the data that we have now gets divided down in significance fairly quickly especially for calculations on non-intermediate patient profiles. Some assumptions that we have made thus far that require further validation include:

- Is it appropriate to use an original patient set window or +/- 50% PSA ? The larger this window becomes the less precise our results become due to the inclusion of patients that may not be a true representation of the case to be modeled. This is especially true for high initial PSA scores.
- Is our model adequately handling PSA initial decline and later bounce behavior?
- Would it be desirable to use a sliding start date window to include more patients of similar PSA and risk in calculations (maybe a window +/- 3 months? Or more)
- Is a minimum start window of +/- 0.2 PSA a good choice or would something large such as +/- .5PSA be better? Should we have a maximum start window limit?
- Should we be categorizing patients by demographic factors such as age?

Doubtless there are many other avenues to explore, which will only become evident as we explore findings and observations which arise from comparative data and adjusting our model/assumptions.