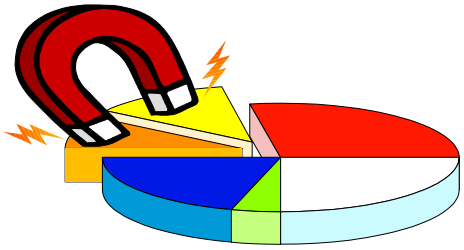
Complex Computation Lab
IOWA STATE UNIVERSITY

Vect Version 1.0 Guided Tour



Hui-Hsien Chou
Denise Mooney
Kazuya Suzuki
Satish Vemula

In recent decades, biomedical researches have faced a new and daunting task of interpreting tremendous amounts of information that have been generated in such areas as genomics, microarrays and proteomics. Specialized data extraction and conversions computer programs are needed to interpret such data for rapid advancement in biomedical research. Vect, essentially, is a visual extraction conversion program that provides biomedical researches with very minimal computer science background a way to generate data extraction and conversion programs without knowing computer programming at all. Vect provides a convenient graphical user interface that allows users to use common point and click methods to arrange data in the way they deem fit and then provides the computer program to generate such results. Vect can be used with virtually any text format generated from computational technologies, making Vect a powerful tool for biomedical scientists.

Please note that this version of Vect is still under development and subject to change. Not all icons are fully functional. More features will be added in the future.

Introduction to Tutorial

This tutorial is designed especially for biomedical researchers to give a basic understanding of the functions of Vect for data extraction and conversion. Users should be able to perform the following tasks upon completion of the tutorial:

- Load a target file
- Select regions to be used in the final format
- Apply rules to the data set
- Arrange the data in a desired format
- Convert the final format to programming code

An Overview of Vect Functionality

Vect (Visual Extraction Conversion Tool) is a program designed to generate Perl programming code that extracts specific data from lengthy files and reports and arrange the data based on user preferences. Auto generation of code is done in various phases. These phases are streamlined, where output from one phase becomes the input of the next phase. The graphical user interface of Vect lets the user step through the phases of loading data files and defining rules to extract specific data and in return generates the Perl code which can run on similar data files. Even though Vect can work on files of any format, semi-structured files (having a predefined outline) will help the user create more meaningful and intuitive rules and results.

The following four phases are involved in the generation of Perl code:

Loading input data: During this phase, multiple input data files (mainly with text and data file extensions) can be loaded into Vect.

Creating rules for extraction: During this phase, users can apply various rules to the input file to extract specific information. There are two types of data extraction: **1) Data-dependent and 2) Rule-dependent**. While data dependent rules are defined on the input

data and provide the easiest way for data extraction, rule-dependent rules are defined based on other rules. These rule-dependent rules provide the most powerful means of manipulating, formatting, filtering and composing various data sets.

Vect offers the greatest potential in future program development with the rule-dependent rules.

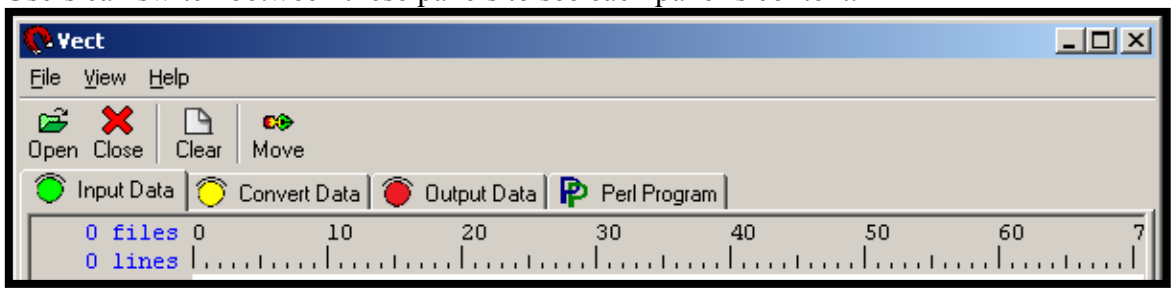
Formatting the output data: During this phase, the rules are organized together to visually output an organized and legible file format. Users can insert headers and footnotes, as well as add text to the existing rules.

Auto generation phase: During this phase Vect generates Perl code based on the previous phases that results in a program that can be run for similar semi-structured files. Users can modify the auto generated Perl code to further customize the output data and its format. Users must keep in mind that the rules defined by the user lead to the Perl code, and not the other way around.

Guided Tour of Vect User Interface

The user interface of Vect consists of four panels:
1) Input Data Panel
2) Convert Data Panel
3) Output Data Panel and
4) Perl Program Panel

Users can switch between these panels to see each panel's content.



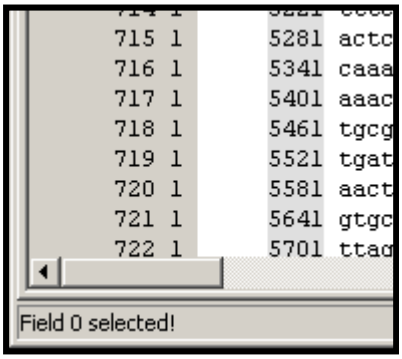
Input Data Panel

Loading of input data files and creation of data-dependent rules are the main functions of this panel. There are four data dependent rules:

1) Field Selection
2) Column Selection
3) Column in Field Selection
4) Forced Column Selection
5) Line Restriction
6) Block Restriction

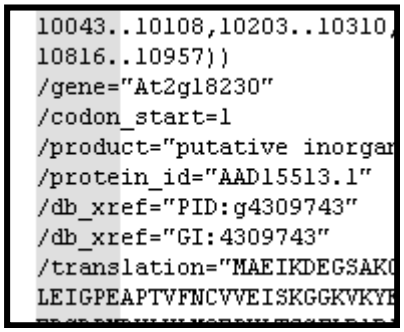
Field Selection

- Vect organizes the data into fields based on word orientation from the left most column, beginning at field 0. A left click on the field will select it. If a word is selected that is located in the third field of a line, then all words and numbers located in the third field will be selected and will be shown by a grey highlighted region. The region can be deselected by again clicking on the grey highlighted region.



Column Selection

- To specify a column of text left-click and drag over the area desired. To deselect the column simply left-click again on the grey highlighted region. Similar to field selection, if column selection is done in one line, all the text in those specified columns would be highlighted in grey.



Column in Field Selection

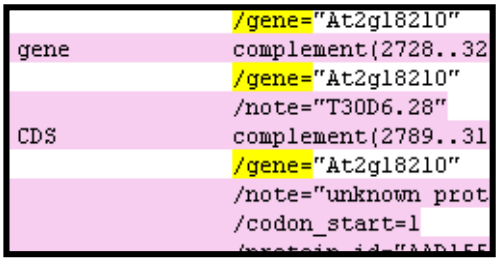
- To specify a fixed column length directly inside a field, left-click inside the field and drag over the desired characters. This feature will be discussed in more detail in later tutorials.

Forced Column Selection

- A column can be selected that is completely inside a field by holding down the shift key while right clicking and dragging over the desired field. This feature will be discussed in more detail in later tutorials.

Line Restriction

- To specify only lines containing a desired word or number, right click and drag over the area desired and select *New Line Selection* in the pull down menu. A yellow box will appear and it is in these lines only that data extraction can occur.

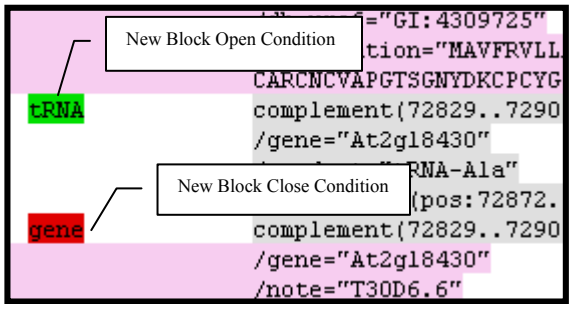


The line selection can be made either *Position Dependent* or *Position Independent*, by right clicking on highlighted boxes and selecting 'Position Dependent or Independent' from the pull down menu. Depending on the option chosen, the restriction will consider the position of the line restriction text as it appears in the line. If *Position Independent* is selected, it does not matter where the line restriction text appears

in the line, all the lines containing the specified text would be selectable. If it is *Position Dependent*, only the lines that have the specified text in the specified location would become selectable.

Block Restriction

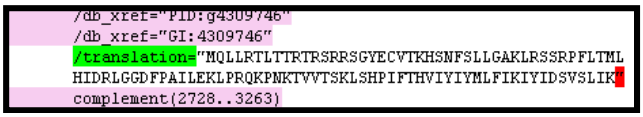
- Similar to Line selection, Block Selection allows restricting the data to be considered for selection. While *Line Selection* allows restricting the selection to specific lines based on a condition, *Block Selection* allows restricting the selection to blocks of data specified by opening and closing conditions. To specify an opening and closing block, right click on a field or right click and drag over the word(s) or number(s) to begin data selection and choose *New Block Open Condition* from the pull down menu. A green box will appear showing the beginning of data selection. Similarly to close data selection, choose *New Block Close Condition* from the pull down menu. A red box will appear showing the end of data selection. To deselect the block conditions right click on green or red regions and choose *Cancellation* from the pull down menu. All pink regions in the input data panel shows the data that cannot be used for data extraction.



Block Selection can be made more useful by using the two sets of options available. Right clicking on the colored block text and selecting from the pull down menu can select these options.

1) Position Dependent & Independent
   This is similar to the option available for *Line selection*. The block condition restriction depends on the position where the *Block Open* or *Block Close* conditions occur in the line, depending on the option chosen.
2) Selection Exclusive & Inclusive
   This option lets the user specify whether the *Block Open* and *Block Close* conditions can appear in the same line. To specify that the line, in which a *block condition* occurs, should not include any more block conditions, right click on the red or green highlighted selection and choose *Selection Exclusive* from the pull down menu. To make the selection inclusive, right click on the red or green highlighted selection and choose *Selection Inclusive* from the pull down menu.

   A sample use of this option can be shown in the following diagram. In this example, the *Block Open* condition is made on '/translation=' and the *Block Close* condition is made on the quotation marks ("). These are then both made *Position Independent* so that the *Block Close* condition does not apply on the quotation (") immediately followed by 'translation=' and because the protein text can vary in length and quotation (") can appear any where in the line.

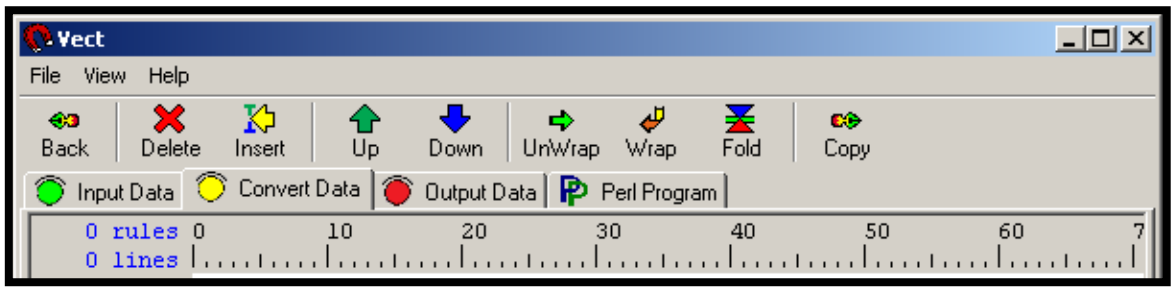


It is not necessary that every *Block Open* condition have a corresponding *Block Close* condition. In case there is no *Block Close* condition, either the *Block Open* condition itself or the end of input will mark the end of block. Nested block conditions can be defined to further restrict the selectable regions.

**While data selection is done with a left-click, for Line and Block Selections a right-click is used.*
**Block or line selections do not select the data in the specified block or line automatically. These selections are used to restrict the selectable areas in the whole input data. Users still have to explicitly select the fields using 'field selection' and 'column selection' and then the selected data will be highlighted in grey.*
** After the Line/Block Selection is defined and the text is selected for extraction, go over the whole input file to make sure that only the text that you desired is being selected. Depending on the text format, you may have to tweak the line/block definitions to extract the specific text that you desire.*

Once the desired data is properly selected, click on the 'Move' button from the icon panel and give your data set a name. The data now appears on the next panel labeled 'Convert Data.' Using the same procedure, multiple data selections can be made and moved over to the Convert Data panel. Naming the data sets meaningfully, depending on the context, will make data extraction process more manageable.

Convert Data Panel



The Convert Data Panel consists only of the rule-dependent rules. There are currently the rules in this panel:

1) Concatenate Data Rule
2) Extract Quoted Data Rule
3) Filter Data Rule
4) Pick Data Rule
5) Merge Data Rule
6) Convert/Reverse Data Rule
7) Translate Data Rule

8) Extract Substrings Rule
9) Write your own Perl Script

Concatenate Data Rule
Allows users to move data in multiple lines into one-line strings based on various levels. Users can also choose to add punctuation or text to separate concatenated lines. There are seven levels, level six being the upper most level (not concatenated at all) and level 0 being concatenated to one string (one line.) The different levels in-between will concatenate based on the level the data is present in. If, for example, the left-panel of your rule is labeled as (3) then concatenating to level three will result in a string for each field present in the original data set. You can determine the number of fields present by the number of stars on the left-panel of your rule. It is a good idea to practice extracting various levels of data and concatenating at various levels.

Extract Quoted Data Rule
The 'Extract Quoted Data' rule allows phrases in between or next to punctuation or other common characters to be easily extracted. A common use is to extract data in between quotation marks. The user will specify between which marks the wanted data is located and Vect will extract that data.

Filter Data Rule
The Filter Data Rule allows specific characters to be extracted (i.e. integers, words, uppercase, lowercase and alphanumeric characters.) A pull down menu allows users to easily pick what characters are wanted.

Pick Data Rule
The Pick Data Rule allows for users to choose which blocks they would like to extract. They can select from choosing every other block, every third, etc or define their own selection.

Merge Data Rule
The Merge Data Rule allows users to combine data from two different rules into one document. A good example is if a user wants to attach the body to a title. The user would select all titles in one rule and all of the body in another rule and then use the Merge Data Rule to combine them.

Convert/Reverse Data Rule
The Convert/Reverse Data Rule allows sequences of data to be reversed or converted in the case of DNA. Users could easily obtain the complementary string of the DNA string that they are editing or flip the string from a three prime to five prime end.

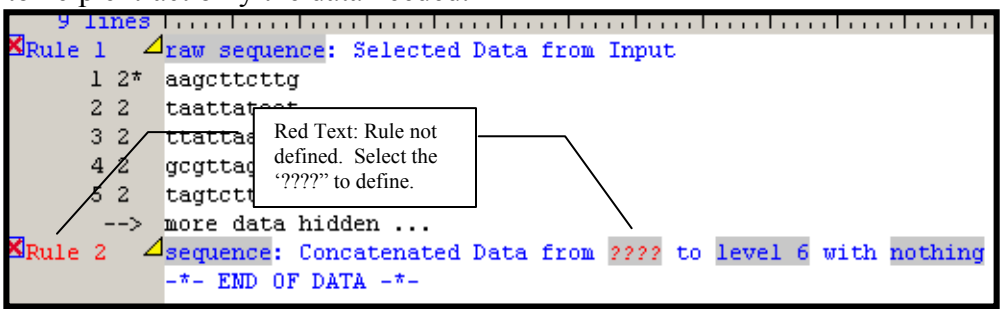
Translate Data Rule
The Translate Data Rule allows for DNA to RNA or RNA to DNA translation. A pull down menu allows users to easily pick what type of translation is needed.

Extract Substrings Rule

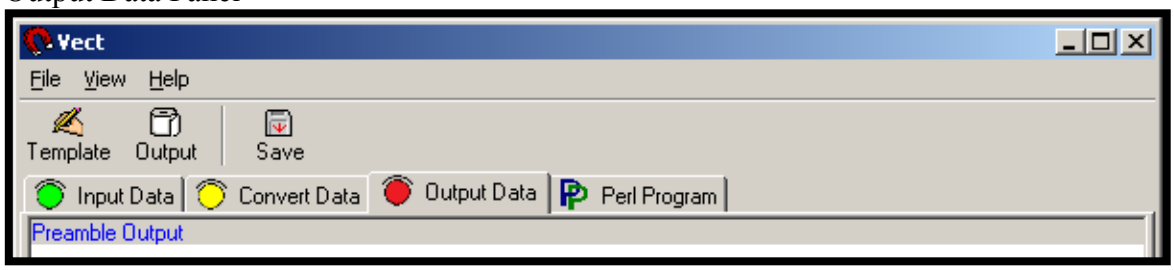
The Extract Substrings Rule allows Vect to find viable information from a different rule to be used in data manipulation. Users could tell Vect where the coordinates are for one set of data and Vect would use these coordinates to grab data.

Write your own Perl Script
The Perl Script Rule allows users to write their own rule, in case their rule is not already given. Users should consult documentation on how to write Perl Script in case they are not familiar with its syntax.

To apply any rule to a data set, simply click on the 'Insert' icon in the icon panel and select the rule you wish to apply. Give the rule a descriptive name and also select the rule you wish to apply it to by selecting the grey highlighted question marks. If the rule has incomplete data, the rule will appear in red on the left-panel. Select the grey boxes with the incomplete rule to specify more information. Change other grey highlighted boxes to help extract only the data needed.

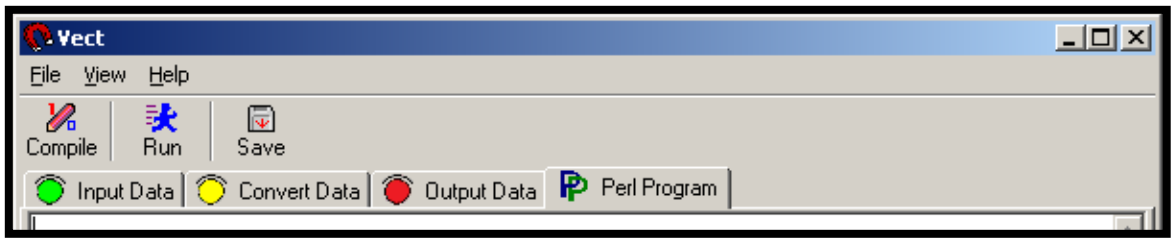


Output Data Panel



The 'Output Data' panel is where a template can be formulated to organize the data collected from extraction. The 'Copy' button is used from the 'Convert Data' panel to copy all wanted rules over. Each rule appears in brackets (< >). These rules should not be edited at this point unless users understand ways to edit. Users may add their own text and format by positioning the <rules> where needed. Users can select the 'Output' icon to visually show how their example will look in that format. If the user would like to make more changes they simply go back to the 'Template' view. Users may add text to the 'Preamble Output' and 'Appendix Output' sections, which will only appear above and below the body.

Perl Program Panel



Users should select the 'Compile' icon to view and save their final Perl program code. At this point, if the user wants to add their own programming touches they may. The data can be run by selected the 'Run' button where a command shell will appear and the data will be formatted based on the Perl code and the file loaded into Vect.

This completes the Guided Tour of Vect. It is best at this point to begin practicing Vect with a sample file tutorial. There are a number of tutorials available for Linux and Windows.

Complex Computation Laboratory
Iowa State University
Denise Mooney
Satish Vemula