User's Manual for FISH
(Fast Identification of Segmental Homology)

version 1.0
released 30 June 2003

Contents

Chapter 1

Getting started

1.1 Introduction

Genomic regions that have descended from a common ancestral region are said to be *segmental homologs*. FISH is software for the fast identification and statistical evaluation of segmental homologs.

The software treats each input genome as a string of *features*. A feature may correspond to one marker, or sometimes to multiple homologous markers that are very close to one another on a contig. Each string of features (whether it is a compete chromosome or some smaller physical or genetic linkage group) is referred to as a *contig*. *Matches* are pairwise homologies among markers, which, along with the position of markers on contigs, constitute the primary input data. FISH finds *blocks* of putative segmental homology by identifying paired regions in which multiple matches occur and where the underlying features are in roughly colinear order. As described more fully below, FISH evaluates the probability of observing each block under a null model in which matches are distributed uniformly at random among features.

Users of FISH should cite the following paper (in which more detail regarding the method can be found):

- Calabrese PP, Chakravarty S, Vision TJ (2003) Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics* 19, i74-i80.

1.2 Contact information

Please send questions, feedback, and bug reports to tjv@bio.unc.edu. The FISH website provides an online form where you may submit your email address should you wish to receive notification about updates to this software.

1.3 Legal information

protection in exchange for a fee.

3. RIGHTS IN DERIVATIVE AND COMBINED WORKS. It is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program. Aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 2 above, provided that you also meet the following conditions: (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date and nature of any change. (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

4. USER'S PROVISION AND/OR DISTRIBUTION OF SOURCE CODE. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the

executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable. If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

5. LICENSE RESTRICTIONS. (a) You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance. (b) You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it. (c) If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through

that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

6. LICENSE CONVEYS TO SUBSEQUENT RECIPIENTS. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. SEVERABILITY. If any portion of this license or any portion of any section of this license is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

8. NO WARRANTY. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program"as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program"as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

1.4 Acknowledgements

Funding for FISH was provided by the USDA-ARS Center for Agricultural Bioinformatics, the Center for Computational Genomics at the Cornell Theory Center, and by National Science Foundation grants DBI-0110069 and DBI-0227314 to TJV and DMS-0102008 to PPC. The authors wish to thank Peter Calabrese for his many contributions to the ideas underlying the software.

1.5 Availability

The latest version of this program, along with documentation and examples, is available from
`http://www.bio.unc.edu/faculty/vision/lab/FISH/`.

FISH is written primarily in C++, together with C style data input/output. The source code is designed to be compatible with all versions of gcc (GNU C++/C compiler). In addition, executables are available for a limited number of platforms. The program may require 32 MB of RAM or more at runtime.

1.5.1 Installing from source code

Installation from source code requires a c++ compiler (the makefile assumes that you have g++). Fish is known to successfully compile with g++ 3.1 and 3.2 (to see what version of g++ you have type “g++ -v”).

The code is known to compile and run on at least some versions of Solaris,

RedHat Linux, Windows and MacOSX. It should compile and run on all systems with g++ installed. For windows environments, MinGW(http://www.mingw.org), provides a g++ compiler that works well. To install, type the following commands.

1. tar -xzvf FISH-1.0.tar.z
2. cd FISH-1.0
3. make all
4. cp bin/fish /usr/local/bin (may require superuser priviliges)

1.5.2 Installing the Windows executable

```
INSTALLATION:
   1. Unzip FISH-1.0-win.zip into "C:\<DESIRED DIRECTORY>".
   2. The executable will be located in
      "C:\<DESIRED DIRECTORY>\FISH-1.0\fish.exe".
For Example:
   Unzip to "C:\Program Files"
   From a command line type: "C:\Program Files\FISH-1.0\fish.exe" -h.
   Don't forget the quotations marks!  This executes fish and shows
   the command line options.

REQUIREMENTS:
   1. 32 MB of RAM.
   2. Operating system:  Known to run on Windows XP,ME, and NT.
```

1.5.3 Installing the MacOSX executable

```
INSTALLATION:
   1. Unzip FISH-1.0-osx.zip.
   2. The executable will be located in "FISH-1.0/fish".
REQUIREMENTS:
   1. 32 MB of RAM.
   2. Operating system:  Known to run on 10.2.x
```

1.5.4 Testing FISH on sample data

You may use the sample files to test that the program is working properly. From a command line window, go to the directory within the distribution where the sample control file, “control.txt”, is located. Type ’fish’, then enter, and within a few seconds FISH should complete the analysis.

Chapter 2

How it works

The following is a concise overview of FISH. A fuller description of the methodology is given in [1], with which we assume the reader is already familiar. There are three steps in the analysis:

Step 1: From markers to features

Step 2: From features to grid

Step 3: From grid to blocks

Each genome to be compared consists of one or more linear contigs, which consist of a sequence of features. A feature is typically one or more closely related protein-coding genes at a particular locus, but may be any entity to which it is possible to ascribe homology to other features. Because FISH considers a contig to be merely an ordering of features, it is in effect treating the distance between adjacent features as one unit.

2.1 Step 1: From markers to features

The names, positions and transcriptional orientations (when known) of the markers are read from a set of *map files*, one map file per contig. Markers within each map file must be ordered according to their physical positions on the contig. FISH assigns each marker a unique identification number, from 1 to the total

number of markers in all the contigs, which serves as a linear coordinate for subsequent calculations. Markers cannot appear more than once on a contig, *i.e* each marker on a contig must be unique; otherwise, an error will result.

Individual homologies between markers are read from a set of *match files*. There is at least one, and no more than two, such files for each pair of contigs. Each match is associated with a score that reflects the strength of that match. Larger scores indicate stronger matches. When two matches between the same markers are present in two different match files (*i.e.* contig 1 vs contig 2, and contig 2 vs, contig 1), the score is taken to be either the maximum or the average of the two according to the user's choice (refer 3).

On the basis of the map positions and matches, FISH performs *detandemization*, in which multiple markers may be collapsed into single features. Two parameters control the detandemization process: MIN_SCORE and MAX_DIST. Markers with coordinates that are separated by MIN_DIST or less will be collapsed into a single feature if they are homologous to each other or to a third marker with a score at least as great as MIN_SCORE.

To calculate the relative order of features in the detandemized contig, each feature inherits the position of the marker having the lowest valued coordinate within the set of markers comprising the feature. The coordinates are then reassigned for features in all contigs from zero to one less than the number of features in the whole dataset. When transcriptional orientations are known, the feature inherits the majority orientation (either '1' or '-1') of the markers that comprise it. For example, if three out of the five markers that comprise a feature have a transcriptional orientation of '1', then the feature inherits that orientation. If there is no majority, then it is assigned a value of zero, for 'unknown'.

The score of a match between two features, when one or both of the features represents multiple markers, is taken, depending on the runtime parameter chosen, to be either the highest score or the average score for any of the pairs of component markers, one from each feature.

2.2 Step 2: From features to grid

In order to identify segmental homologies, FISH computes a grid for each pair of contigs. *Points* in the grid represent matches between pairs of features. The i and

j coordinates of a point are taken to be the positions of the features within their respective contigs. Each position in the grid, whether or not a point is present, is called as a *cell*.

2.2.1 Removing low ranking matches

The number of matches for each feature in the final grid can be limited by an adjustable parameter, TOP_HITS. If a match is not one of the highest TOP_HITS scoring matches for at least one of the features, then it is deleted. This step is necessary to removes some of the noise resulting from large gene families.

2.2.2 Numbering of points

Following this, each point in the grid is given a unique identifying number from zero to one less than the total number of points. These numbers are assigned to points in sorted hierarchical order, first by one coordinate and then by the other. Cells for which $i \geq j$ are not included.

2.2.3 Enforcing symmetry

The above preprocessing steps have ensured that symmetry has been enforced in the grid, such that the match between features i and j has the same score as the match between features j and i. Thus, we only need to consider the half of the grid containing i versus j in searching for blocks.

2.2.4 Defining the neighborhood size

Blocks consist of sequences of points in which each point is within the neighborhood of the previous one. FISH calculates the size of the *neighborhood* for all points based on the total number of points m and cells n in the grid. The probability of a cell containing a point is $h = m/n$. From this, and from an adjustable parameter T (see 3.1), FISH calculates the size of the neighborhood. In its current version, FISH measures distance between two points (X_i, Y_i) and (X_j, Y_j) using the Manhattan distance, $d = |X_i - X_j| + |Y_j - Y_i|$. In order to be

considered neighbors, two points must be closer than

$$d_T = \frac{1}{2} + \sqrt{\frac{\log(1-T)}{\log(1-h)}} + \frac{1}{4} \tag{2.1}$$

2.3 Step 3: From grid to blocks

FISH uses a dynamic programming algorithm to identify all maximally extended blocks. For a detailed description of the algorithm, see [1].

2.3.1 Choosing among multiple neighbors

It can happen that a point may be in the neighborhood of more than one other point. In such a case, it is necessary to specify rules for which point will be chosen to extend the block. To do this, FISH ranks the cells within each neighborhood and chooses that neighbor having the highest rank, using an *ad hoc* formula that gives preference to points that preserve the colinear order of matching features along the two contigs. The rank r for each cell in the neighborhood of the candidate point is

$$r = w\frac{1/d_c}{\sum_1^n (1/d_c)} \tag{2.2}$$

where n is the number of cells in the point's neighborhood, d_c is the distance of the cell from the point under consideration and w is the weight. The weight

$$w = \begin{cases} 1 & \text{if the point is the first in the block,} \\ 0.25 & \text{if an inversion is implied by the extension,} \\ 0.75 & \text{otherwise.} \end{cases} \tag{2.3}$$

When multiple points have equal rank, the point is chosen that has the smallest value of $|(|X_i - X_j|) - (|Y_j - Y_i|)|$. This favors points close to the diagonal.

2.3.2 Overlapping blocks

FISH processes blocks in such a way as to ensure that only non-overlapping blocks are obtained. If a point could be included in multiple blocks, one of the

blocks with highest score is chosen. The point is then made unavailable for inclusion in other blocks.

Chapter 3

Parameters

A number of parameters may be adjusted at the command-line when FISH is invoked. They may also be reviewed from the command line by invoking FISH with the help switch:

```
# fish -h
```

The adjustable parameters are listed here and described in more detail below.

parameter	switch	default	acceptable values
PRINT_BLOCKS	-b	off	valid file name
PRINT_B_SIMPLE	-B	off	valid file name
MIN_BLOCK_SIZE	-m	3	integer greater than 1
PRINT_CONTIGS	-C	off	valid filename
CONTROL_FILE	-f	control.txt	valid filename
PRINT_GRIDS	-g	off	valid filename
TOP_HITS	-H	5	integer greater than 0
MAX_DIST	-D	10	integer greater than 0
BLOCK_PROB	-p	0.001	real between 0 and 1
QUIET_MODE	-q	off	none
MIN_SCORE	-S	200	real
TIMING	-t	off	none
AVG_SCORE	-A	Max_Score	none
T	-T	0.05	real between 0 and 1
DETANDEMIZE	-off	on	none

3.1 Algorithmic parameters

A number of the adjustable parameters control the way in which FISH finds blocks. Altering these parameters can substantially influence the results.

MIN_BLOCK_SIZE The minimum number of points needed to define a homologous block.

TOP_HITS Matches that are not in among the TOP_HITS scoring matches for either component feature will be discarded.

MAX_DIST The maximum number of intervening markers allowed between markers in t he same detandemized feature.

T The probability of having one or more neighbors within a distance less than or equal to than d_T under the assumption that each cell contains a point with probability h. Values of T closer to zero will result in blocks with more closely spaced points. This is roughly analogous to the gap parameter in local sequence alignment.

MIN_SCORE Matches with scores less than MIN_SCORE will not be considered by FISH despite being listed in a match file.

AVG_SCORE This option enforces symmetry between two matches (*e.g.* 1 versus 2 and 2 versus 1) by taking an average of the scores. If this averge is less than the MIN_SCORE, the match will be discarded. The default behavior is to choose the larger of the two scores.

DETANDEMIZE This allows the user to turn detandemization off.

3.2 Input and output parameters

The remaining parameters control the input to and output from FISH.

QUIET_MODE If selected, FISH will suppress the default screen output (see 4.2.1). This switch takes no arguments.

TIMING If selected, FISH will report the number of seconds elapsed between various execution time points. This switch takes no arguments.

CONTROL_FILE Specifies the name of the file containing the instructions for FISH to read particular contig and match files 4.1.1. If not specified at the command-line, FISH will search for a file named "control.txt" and use that, if available.

PRINT_CONTIGS Specifies the name of the file to which FISH writes contig output (see 4.2.2). Unless selected, no file will be written.

PRINT_GRIDS Specifies the name of the file to which FISH writes grid output (see 4.2.3). Unless selected, no file will be written.

PRINT_BLOCKS Specifies the name of the file to which FISH writes block output in human-friendly format (see 4.2.4). Unless selected, no file will be written.

PRINT_B_SIMPLE Specifies the name of the file to which FISH writes an alternative format for the block output that is more computer friendly (see 4.2.4). Unless selected, no file will be written.

BLOCK_PROB This specifies a value for the confidence level of a block. The confidence level is calculated as the expected number of blocks with n points divided by the observed number of blocks with n points. If the observed number does not sufficiently exceed the expected number (under the null model), then blocks of that size are flagged in the summary output. Some of these blocks may well be spurious.

Chapter 4

File formats

4.1 Input files

There are three types of input files: control files, map files and match files. The control file tells FISH the location of the map and the match files.

4.1.1 Control file

A control file is used to specify the paths of the map and match files. The format of the control file can be seen in the sample provided. It is divided into two sections, one for map files and one for match files.

At a minimum, one map file and its associated match file are required. To compare two different contigs, map and match files are required for each. In addition, one or two match files must be specified that list the matches between features in contig 1 *vs.* 2, and/or 2 *vs.* 1.

A list of the map files is preceded by the optional token '-maps' followed by a newline character, followed by a separate line for each contig that is to be analyzed. Each of those lines contains an integer, followed by whitespace, followed by a valid filename path, followed by a newline character. The integer is taken to be the name of a contig and the filename path tells FISH where to find the map file for that contig.

A list of the match files is preceded with the optional token '-matches' followed

by a newline character, followed by a separate line for each pair of contigs to be analyzed. Each of those lines contains an integer, followed by whitespace, followed by an integer, followed by whitespace, followed by a valid filename path. The two integers specify the contigs for the match file. Note that there must be at least one comparison in which the integer of the first contig is less than the integer of the second contig. For example if 1 versus 2 does not exist, than 2 versus 1 is ignored. So if there is only one match file for a comparison between two contigs then that comparison must be written as 1 versus 2.

4.1.2 Map files

Each map file lists the names and transcriptional orientation (if known) of all the markers on one contig. For an example, see Section 5. Markers must be listed in positional order. Each line contains a unique marker name, followed by a tab character, followed by the transcriptional orientation. Marker names can be character string up to a maximum length of 100 that does not include whitespace or the '&' character. (This value can be modified in the file "fish.cpp" by changing the value of MAXBUFFER and recompiling). Marker orientation is taken relative to the contig as a whole. Thus, if marker 1 is coded on the same strand as marker 2, then both markers should be assigned the same direction. Orientation can take values '-1', '0' and '1'. Zero is used for markers of unknown orientation.

4.1.3 Match files

Each match file lists all the homologies between markers in a pair of contigs. For an example, see Section 5. There are three columns in the matchfile. The first two columns may be strings, but the third column needs to be an integer. The first and second columns contain the gene names from the first and second contig respectively. The integer in the third column must be some measure of the strength of the match between the two markers. FISH assumes that the larger the integer, the stronger the match. If information is not available on the strengths of the matches between genes, one can, for example, assign all scores to be zero, and set MIN_SCORE to be zero at runtime. All self matches are discarded.

4.2 Output

FISH can generate a variety of different output streams depending on the user's selections. FISH can also report the time spent at various stages in the analysis. This option can be selected by using the -t switch (see Section 3).

4.2.1 Screen output

Unless the QUIET_MODE has been selected, FISH will report results from each stage of the analysis to standard output (STDOUT). See Section 5 for an excerpt from the sample file "fish_out.txt". The STDOUT stream may be directed to a file, rather than to the terminal, by using the > redirection operator, like so

```
# fish [options]> myfile.txt
```

At the beginning of the process, FISH sends STDOUT a brief title message, including the current version number, the date and time at which the process was initiated, and the control file name.

Processing contigs

FISH displays the values of the two relevant parameters for the contig processing stage analysis: MIN_SCORE and MAX_DIST. Once detandemization is complete, a table is written that displays the number of markers and features within each contig. Detailed information on which markers compose which features is written to a file if the PRINT_CONTIGS parameter has been selected.

Processing matches

FISH displays the values of the relevant parameters of the match processing stage, MIN_SCORE and TOP_HITS. Note that the current and previous value for for MIN_SCORE must always agree. Once the calculations are complete, FISH displays the number of points and the number of cells for each pair of contigs in the analysis. In addition, the coordinates of the points in the detandemized dataset are written to a file if PRINT_GRIDS has been selected.

Processing blocks

FISH first displays the key adjustable parameter of the block finding stage, T. Following this is a series of numbers which depend on both the dataset and the adjustable parameters:

- N, the total number of points in the grid
- F, the total number of features in the dataset
- the total number of cells in the grid, calculated as $F(F-1)/2$
- h, the proportion of cells in the grid that contain a point
- d_T, the neighborhood size within which there is only a T probability of encountering another point if points in the grid are distributed uniformly at random (see Equation 2.1)
- MIN_BLOCK_SIZE, the number of points in the smallest block to be reported.

Once the blocks have been identified, FISH will also display the total number of blocks.

The length distribution of the blocks is displayed as a vertical histogram. To the right of each bar is listed the number of blocks of length L, as measured by the number of points. If the number of blocks is greater than 50, the true amount is not represented by '*' characters and the block frequency is preceded by '...'. To the left of the histogram is shown the corresponding p-value for observing a block of length L (see Section 4.2.4), and the expected number, under the null model. If the proportion of expected to observed blocks of length L is greater than BLOCK_PROB, then the frequency of observed blocks is followed by an exclamation mark. This indicates that blocks of this length have not been observed sufficiently more frequently than expected, and so the individual blocks may not be biologically meaningful.

4.2.2 Contig file

If the PRINT_CONTIGS option is selected, FISH will output a contig file with the specified name. See Section 5 for an example. The contig file allows

translation between feature numbers and marker names.

The top of the contig file displays the time at which the process was initiated. This is followed by the parameter list, which includes the name of the control file and the values of MIN_SCORE and MAX_DIST. This is followed by a table showing the number of markers and features in each contig. Finally, a table is presented in which, for each feature, the contig, the composite direction and the names of the markers (from the map file) associated with that feature are shown.

4.2.3 Grid file

If the PRINT_GRIDS option is selected, FISH will output a grid file with the specified name. See Section 5 for an example. The grid file allows translation between points and features as well a summary of the parameters and results for the match processing step.

The top of the grid file displays the time at which the process was initiated. This is followed by the parameter list, which includes the name of the control file and the values of MIN_SCORE and TOP_HITS. This is followed by a table showing the number of points and cells for each pair of contigs. Finally, the points are listed. For each point, the associated contigs, the associated features, and the composite score are shown. See Section 4.2.4 for an explanation of how the score is derived.

4.2.4 Block file

If the PRINT_BLOCKS option is selected, FISH will output a block file with the specified name. See Section 5 for an example.

The top of the block file displays the time at which the process was initiated and the control file used. This is followed by the parameter list. This list, similar to the one output to STDOUT (see 4.2.1) includes the name of the control file, the value of the threshold paremeter T, the total number of points in the grid, the total number of features in the dataset, the total number of cells in the grid, the proportion of cells that contain a point h, the size of the neighborhood d_T, the minimum block size to be reported, and the total number of blocks found in the dataset.

This is followed by individual tables for each block. Within each, there is an upper panel that lists the number of points and the associated contigs. And there is a lower panel that lists each point in the block in sequence. For each, the table shows the number of the point, the distance from the previous point, the names of the markers for the associated features (in the format markers for feature 1 markers for feature 2), and the product of the orientation of the point with the previous point in the block. This product may take a value of -1, 0 or 1. If values switch frequently between -1 and 1, then transcriptional orientation is not well-conserved within that block, and the user may wish to regard this block with suspicion. Zero indicates 'unknown'.

At the bottom of the blocks file, the number of blocks seen between each pair of contigs is reported. Below that, the length distribution of blocks is reported. For blocks with a given number of points, the observed and expected numbers of blocks are shown followed by the p-value for blocks of that length. The p-value is calculated as described in ([1])

$$p = 1 - e^{-rcp_u} \tag{4.1}$$

where rc is the total number of cells in the grid, $p_u = h(nh)^{k-1}$, and n is the number of cells in a neighborhood of size d_T.

4.2.5 Simple block file

FISH provides the option of printing a more computer readable block file by choosing the PRINT_B_SIMPLE option. The first two lines report the number of blocks and the total number of points in all blocks, respectively. Each subsequent line reports the output for one block. The first number in each line is the number of points in that block. This is followed by comma-delimited list of the points themselves. Each block is terminated with an asterisk.

Chapter 5

An example

The following truncated input and output files come from a re-analysis of the segmental homology within and among the five chromosomes of *Arabidopsis thaliana* [1, 2] All the genes that received AtXgXXXXX codes in the TIGR version 3.0 release of the genome annotation have been included in this dataset. Assuming that one is in the same directory as the example input files, one can reproduce this session by invoking FISH with the following options:

```
# fish -b blocks.txt
     -g grids.txt
     -C contigs.txt
     -B simpleblocks.txt
     -t > fish_out.txt
```

An explanation of the content of these files is provided in chapter 4.

5.1 A control input file

Shown is the complete “control.txt” file. The filenames may be entered as paths. For example, if the map and match files were in a subdirectory named “data”, the names would read, *e.g.* “datamap1.txt” on a Unix system and would read “data\map1.txt” on a Windows system.

```
-maps
```

```
1     data/map1.txt
2     data/map2.txt
3     data/map3.txt
4     data/map4.txt
5     data/map5.txt
-matches
1     1     data/match1v1.txt
2     1     data/match2v1.txt
3     1     data/match3v1.txt
4     1     data/match4v1.txt
5     1     data/match5v1.txt
1     2     data/match1v2.txt
2     2     data/match2v2.txt
3     2     data/match3v2.txt
4     2     data/match4v2.txt
5     2     data/match5v2.txt
1     3     data/match1v3.txt
2     3     data/match2v3.txt
3     3     data/match3v3.txt
4     3     data/match4v3.txt
5     3     data/match5v3.txt
1     4     data/match1v4.txt
2     4     data/match2v4.txt
3     4     data/match3v4.txt
4     4     data/match4v4.txt
5     4     data/match5v4.txt
1     5     data/match1v5.txt
2     5     data/match2v5.txt
3     5     data/match3v5.txt
4     5     data/match4v5.txt
5     5     data/match5v5.txt
```

5.2 A map input file

Shown are the first ten lines from the chromosome 1 map file (“map.txt”). The first column is the marker (in this case, gene) name, the second column is the

transcriptional orientation.

```
At1g01010      1
At1g01020     -1
At1g01030     -1
At1g01040      1
At1g01050     -1
At1g01060     -1
At1g01070     -1
At1g01080     -1
At1g01090     -1
At1g01100     -1
```

5.3 A match input file

Shown is a sample of ten lines from the chromosome 1 vs. chromosome 1 match file ("match1v1.txt"). The scores shown are bit scores from TBLASTX. Note that with a MIN_SCORE of 200, all but the last match shown would be discarded.

```
At1g01030     At1g25560     112
At1g01030     At1g50680      73
At1g01030     At1g51120      72
At1g01030     At1g35240      63
At1g01030     At1g35520      60
At1g01030     At1g77850      58
At1g01030     At1g34310      58
At1g01060     At1g18330     133
At1g01060     At1g01520     109
At1g01070     At1g11450     313
```

5.4 An example of captured STDOUT

By using the redirection operator, the output that FISH writes by default to the terminal is captured to a text file. In the sample output, this file is named "fish_out.txt"

```
Fri Jun 27 11:15:47 2003
FISH v1.0
FISH is copyright 2003, University of North Carolina at Chapel Hill
Authors: Sugata Chakravarty and Todd J. Vision

Control File = control.txt

processing contigs
        min_score = 200
        max_dist = 10

    contig   msrkers  features
         1      6494      5913
         2      4038      3711
         3      5221      4777
         4      3825      3467
         5      5888      5453

writing contig output to contigs.txt


processing matches
        min_score = 200
        top_hits = 5

      contig1   contig2   points       cells

         1         1       2143     17478828
         1         2       2018     21943144
         1         3       2088     28246400
         1         4       1538     20500372
         1         5       1986     32243588
         2         2        751      6883905
         2         3       1460     17727448
         2         4       1239     12866037
         2         5       1164     20236084
         3         3        849     11407476
         3         4       1300     16561859
```

```
         3          5       1977     26048980
         4          4        711      6008311
         4          5       1551     18905552
         5          5       1293     14864878

writing grid output to grids.txt


processing blocks
        T = 0.05
        total_points = 22068
        total_features = 23321
        total_cells = 271922816
        probability h = 0.000081
        d_T = 25.640781
        min_edges = 2
        block_prob = 0.00100
        total_blocks = 508

   contig1   contig2   blocks
        1         1       50
        1         2       52
        1         3       43
        1         4       34
        1         5       25
        2         2       27
        2         3       48
        2         4       48
        2         5       18
        3         3       15
        3         4       25
        3         5       53
        4         4       16
        4         5       33
        5         5       21

writing output to blocks.txt
```

```
block statistics

points  p-value    frequency
3       1.00e+00        |****************************************
**...243!
4       9.22e-01        |****************************************
***...83!
5       1.17e-01        |*********************************35!
6       6.02e-03        |********************21
7       2.94e-04        |**********************22
8       1.43e-05        |************13
9       7.15e-07        |***********12
10      5.96e-08        |*****5
11      0.00e+00        |***********12
12      0.00e+00        |**2
13      0.00e+00        |****4
14      0.00e+00        |*******7
15      0.00e+00        |*****5
16      0.00e+00        |***3
17      0.00e+00        |***3
18      0.00e+00        |**2
19      0.00e+00        |****4
20      0.00e+00        |***3
21      0.00e+00        |***3
22      0.00e+00        |**2
23      0.00e+00        |****4
24      0.00e+00        |*1
25      0.00e+00        |*1
26      0.00e+00        |****4
27      0.00e+00        |*1
28      0.00e+00        |**2
29      0.00e+00        |*1
30      0.00e+00        |**2
31      0.00e+00        |**2
34      0.00e+00        |*1
40      0.00e+00        |*1
44      0.00e+00        |*1
60      0.00e+00        |*1
```

```
66      0.00e+00        |*1
79      0.00e+00        |*1
                        |----|----|----|----|----|----|----|----|--
--|----|
                       0         10        20        30        40
       50
```

5.5 A contig output file

The opening lines of the file “contigs.txt” are shown here.

```
Fri Jun 27 11:15:44 2003
FISH v1.0
FISH is copyright (c)2003, University of North Carolina
                           at Chapel Hill
Authors: Sugata Chakravarty and Todd J. Vision

-parameters
       MIN_SCORE = 200
       MAX_DIST = 10

-contigs

    contig   markers  features
         1      6494      5913
         2      4038      3711
         3      5221      4777
         4      3825      3467
         5      5888      5453



-features

   feature    contig orientation     genes
         0         1           1 At1g01010
```

```
 1         1          -1 At1g01020
 2         1          -1 At1g01030
 3         1           1 At1g01040
 4         1          -1 At1g01050
 5         1          -1 At1g01060
 6         1          -1 At1g01070
 7         1          -1 At1g01080
 8         1          -1 At1g01090
 9         1          -1 At1g01100
10         1           1 At1g01110
11         1          -1 At1g01120
12         1          -1 At1g01130
13         1          -1 At1g01140
14         1          -1 At1g01150
15         1           1 At1g01160
16         1          -1 At1g01170
17         1           1 At1g01180
18         1          -1 At1g01190
19         1          -1 At1g01200
20         1           1 At1g01210
21         1           1 At1g01220
22         1           1 At1g01230
23         1           1 At1g01240
 .         .           .     .
 .         .           .     .
 .         .           .     .
```

5.6 A grids output file

The opening lines of the file “grids.txt” are shown here.

```
Fri Jun 27 11:15:46 2003
FISH v1.0
FISH is copyright (c)2003, University of North Carolina
                           at Chapel Hill
Authors: Sugata Chakravarty and Todd J. Vision
```

```
Control File = control.txt

-parameters
        MIN_SCORE = 200
        TOP_HITS = 5

-subgrids
       contig1  contig2   points     cells
         1          1       2143    17478828
         1          2       2018    21943144
         1          3       2088    28246400
         1          4       1538    20500372
         1          5       1986    32243588
         2          2        751     6883905
         2          3       1460    17727448
         2          4       1239    12866037
         2          5       1164    20236084
         3          3        849    11407476
         3          4       1300    16561859
         3          5       1977    26048980
         4          4        711     6008311
         4          5       1551    18905552
         5          5       1293    14864878


-points

       point     contig1   contig2      feat1     feat2      score
         0          1         1          6       954        328
         1          1         1         11       300        395
         2          1         1         11       616        237
         3          1         1         11      1668        665
         4          1         1         11      2146        580
         5          1         1         11      4808        582
         6          1         1         11      5050        326
         7          1         1         13      2412        290
         8          1         1         13      2499        303
```

```
 9          1          1          13        3263         401
10          1          1          18        5288         207
11          1          1          30        1288         588
12          1          1          31        1355         549
13          1          1          31        1699         553
14          1          1          41        4440         242
15          1          1          49        4531         273
 .          .          .           .          .           .
 .          .          .           .          .           .
 .          .          .           .          .           .
```

5.7 A blocks file

Selected lines from the file “blocks.txt” are shown here.

```
Fri Jun 27 11:15:47 2003
FISH v1.0
FISH is copyright (c)2003, University of North Carolina
                           at Chapel Hill
Authors: Sugata Chakravarty and Todd J. Vision

Control File = control.txt

-parameters
        T = 0.05
        total points = 22068
        total_features = 23321
        total_cells = 2.71923e+08
        h = 8.11554e-05
        d_T = 25.6408
        Min_Block_Size = 3
        block_prob = 0.001
        total_blocks = 508


-block 0
```

```
    points   contig1   contig2
       79        1        1

       point   dist      markers       orientation
       1040     0       {At1g19320 }{At1g75040 }  1
       1041     2       {At1g19330 }{At1g75060 }  1
       1042     3       {At1g19350 }{At1g75080 }  1
       1043     4       {At1g19360 }{At1g75110 At1g75120 }  1
       1044     3       {At1g19370 }{At1g75140 }  1
       1050     7       {At1g19400 }{At1g75180 }  1
       1056     8       {At1g19450 }{At1g75220 }  1
       1057     3       {At1g19480 }{At1g75230 }  1
       1058     11      {At1g19550 At1g19570 }{At1g75270 }  1
       1061     15      {At1g19650 }{At1g75370 }  1
       1062     2       {At1g19660 }{At1g75380 }  1
       1063     4       {At1g19680 }{At1g75400 }  1
       1064    3        {At1g19700 }{At1g75410 }  1
         .      .                    .                .
         .      .                    .                .
         .      .                    .                .


-block 1
    points   contig1   contig2
       30        1        1

        point  dist     markers orientation
        1246    0       {At1g22920 }{At1g71230 }  -1
        1249    12      {At1g22970 At1g22980 }{At1g71150 }  -1
        1250    14      {At1g23010 }{At1g71040 }  -1
        1251    4       {At1g23030 }{At1g71020 }  -1
        1252    13      {At1g23080 }{At1g70940 }  -1
        1256    7       {At1g23110 }{At1g70900 }  -1
        1258    8       {At1g23140 }{At1g70800 At1g70810 }  -1
        1262    6       {At1g23170 }{At1g70770 }  -1
        1263    6       {At1g23190 }{At1g70730 }  -1
        1265    4       {At1g23210 }{At1g70710 }  -1
        1266    6       {At1g23240 }{At1g70670 At1g70680 }  -1
```

```
        1267    3       {At1g23260 }{At1g70660 }  -1
        1268    9       {At1g23290 }{At1g70600 }  -1
         .      .                  .                  .
         .      .                  .                  .
         .      .                  .                  .
  .
  .
  .
-block 507
    points   contig1   contig2
        3        5        5

        point  dist     markers orientation
        21978   0       {AT5g53950 }{AT5g61430 }  1
        21982   23      {AT5g54230 }{AT5g61420 }  1
        21988   21      {AT5g54380 }{AT5g61350 }  -1


-by contig
   contig1   contig2   blocks
        1        1       50
        1        2       52
        1        3       43
        1        4       34
        1        5       25
        2        2       27
        2        3       48
        2        4       48
        2        5       18
        3        3       15
        3        4       25
        3        5       53
        4        4       16
        4        5       33
        5        5       21

-by size
    points       obs       exp                p
```

3	243	5.23e+01	1.00e+00
4	83	2.55e+00	9.22e-01
5	35	1.24e-01	1.17e-01
6	21	6.04e-03	6.02e-03
7	22	2.94e-04	2.94e-04
8	13	1.43e-05	1.43e-05
9	12	6.97e-07	7.15e-07
10	5	3.40e-08	5.96e-08
11	12	1.65e-09	0.00e+00
12	2	8.05e-11	0.00e+00
.	.	.	.
.	.	.	.
.	.	.	.
44	1	0.00e+00	0.00e+00
60	1	0.00e+00	0.00e+00
66	1	0.00e+00	0.00e+00
79	1	0.00e+00	0.00e+00

Bibliography

[1] Calabrese PP, Chakravarty S, Vision TJ (2003) Fast identification and statistical evaluation of segmental homologies in comparative maps. *Bioinformatics* **19**, i74-i80.

[2] TJ Vision, DG Brown, SD Tanksley (2000) The origins of genomic duplications in *Arabidopsis*. *Science* **290**: 2114-2117.