

CmpSpots - 2D Database Spot Pairing pre-alpha version

Welcome To CmpSpots

<http://www.lecb.ncifcrf.gov/Open2Dprot/CmpSpots/>
 and
 (<http://open2dprot.sourceforge.net/CmpSpots> mirror)

Note: This pre-alpha level software is under construction and will change. Not all functionality described is fully implemented or fully debugged. This documentation will undergo revision as the program develops.

Introduction

CmpSpots is a Open2Dprot project Java 2D spot data pairing program for finding paired spots between two samples. A composite quantified protein expression database consisting of protein expression per (spot) for N samples may be constructed from N sample spot lists. CmpSpots is a program to aid in the construction of this type of database by pairing corresponding spots between spot lists based on their 2D (x,y) locations relative to one another.

The spot list data consists of quantified spot with location (x,y) location that varies between samples. Such data could have been created using image segmentation (e.g., 2D electrophoretic gels), derived from non-image data using clustering methods (e.g., 2D LC-MS) to define peptide "spots", or other methods. For example, if the samples are 2D electrophoretic gels, when we refer to 'sample' we imply gel samples and the spot list of the gel could have been produced by the Open2Dprot Seg2Dgel spot segmenter and quantification program. Open2Dprot is being designed to handle data from a variety of data sources.

It is a step [4] module in pipeline analysis for the Open2Dprot project. Spot pairing is performed on data from a reference sample and another sample to be paired with that reference sample. The method uses a set of landmarks defined between the reference sample and the other sample. Landmarks could be created in a [Future] step [3] module in pipeline analysis operation.

The program may be run either interactively (-gui) with a graphical user interface (GUI) or under an OS shell command to implement batch (-nogui). In the former, after the segmentation is finished, the user has the option of interactively viewing the spot pairing. They may adjust parameters and re-pair the data.

You can currently download the pre-alpha version and install it on your computer. Currently, CmpSpots is hardwired to start with the demo samples and with the -gui switch. The remainder of this home page contains links to some screen shots of the interactive GUI. The Web site contains some initial (rudimentary) documentation.

See the Reference Manual for details. You read about downloading and installing the program on your computer. The source code will be put onto open2dprot.sourceforge.net when it is a bit more stable - currently it undergoing major refactoring.

Examples - samples of screen shots

To give the flavor of running the spot pairing, we provide a few screen shots of the graphical user interfaces and some results. You can these images in the list below or view all of the screen shots in a single Web page.

- Initial user interface Report Window
- Initial command line options tool

Please contact us with suggestions and comments. If you make interesting changes in the source code, please send us a copy and describe your changes so we can merge them in the released version.

*Contact
us*

CmpSpots is a contributed program available at
Open2Dprot.sourceforge.net/CmpSpots
Powered by SourceForge Logo

Revised:
11/07/2005

CmpSpots Table Of Contents

Table of Contents

CmpSpots

- Home
- Introduction
- Overview (PDF) (2) (6)
- Reference Manual(PDF)
- Contributors

- Reference manual
- License
- PDFs
- Tutorials
- Demos & screen shots
- Version
- Downloads
- Status
- Revision history
- FAQ
- References

Source code

- Project summary
- CVS access
- Browse CVS
- CmpSpots Javadocs
- O2Plib Javadocs
- Files mirror

Other Web resources

Hosted at Open2Dprot
open2dprot.sourceforge.net

Powered by SourceForge Logo

Contact us
Revised: 05/19/2006

CmpSpots Program Contributors

CmpSpots Program Contributors

The current CmpSpots program is the result of a collaborative effort with contributions from various groups. The current Java program is partly based based on some of the code refactored from the C/Unix GELLAB-II 2D gel spot pairing program (cmpgl2, 1993 version). That in turn had been refactored from a DEC-10/SAIL language version. The new Java version also includes code from the open source Seg2Dgel, Flicker, and MAExplorer programs.

Many individuals have been involved over the years in its development both as direct collaborators running 2D gel experiments as well as in offering suggestions for improvement including: Lew Lipkin, Eric Lester, Jim Myrick, Mary Robinson, Carl Merrill, Heinz Busse, Peter Sonderegger, Pete Rogan, Ye Chung Wu, Steve Aley, Mike Alley, Rob Ashmore, Kyle Upton, Jeff Orr, Mort Schultz, George Carman, Mark Miller, Arthur Olson, Norman and Leigh Anderson, Sam Hanash, Jian Li, and many

others in the electrophoresis community. Published descriptions of the algorithms for the GELLAB-II program on the Open2Dprot Web site and are in the References section of this Web site.

The core group developing the CmpSpots program source currently consists of

- * Peter Lemkin
- * Greg Thornwall
- * Jai Evans

Contact
us

CmpSpots is a contributed program available at
open2dprot.sourceforge.net/CmpSpots

Revised:
04/20/2005

CmpSpots Table Of Contents

Table of Contents

CmpSpots

- Home
- Introduction
- Overview (PDF) (2) (6)
- Reference Manual(PDF)
- Contributoyrs

Reference manual

1. Introduction
 - Project folder
 - Input spot list files
 - Paired spot-list file
 - Paired-spot viewer
2. Algorithm
 - Primary pairing
 - Secondary pairing
3. Running CmpSpots
 - Computing windows
 - Local file folders
 - Command line args
 - Options window
 - Updating CmpSpots
 - Increasing memory
4. Command window
5. Pull-down menus
 - Menu notation
 - File menu
 - Edit menu
 - View menu
 - Pair menu
 - Help menu
6. Install CmpSpots
 - Running Seg2Dgel
 - Requirements
 - Files downloaded
7. List of switches
 - Switch examples
 - List of -debug bits
8. Demonstrations
 - Screen shots list

All screen shots
Report output
Output file - XML
Output file - SPF
Batch processing
9. References

License
PDFs
Tutorials
Demos
Version
Downloads
Status
Revision history
FAQ
References

Source code

Project summary
CVS access
Browse CVS
CmpSpots Javadocs
O2Plib Javadocs
Files mirror

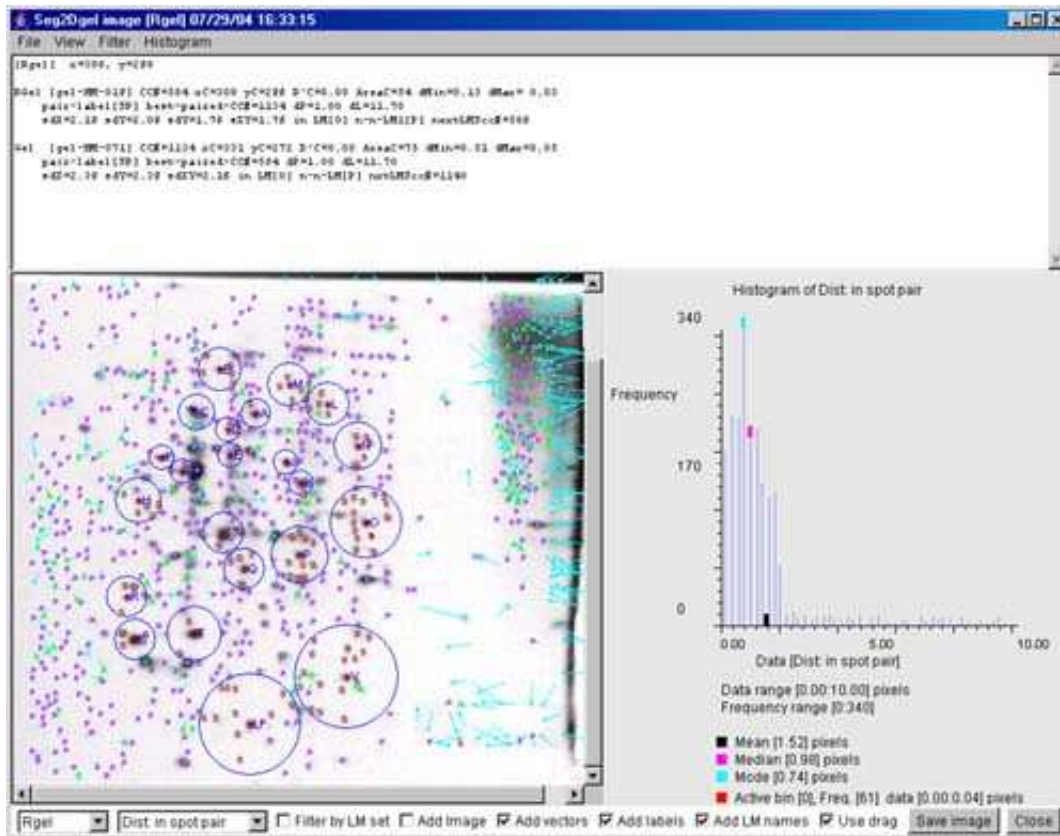
Other Web resources

Hosted at Open2Dprot
open2dprot.sourceforge.net

Powered by SourceForge Logo

Contact us
Revised: 05/19/2006

CmpSpots Reference Manual



CmpSpots Reference Manual

The following description will tell you what the options are for the CmpSpots and how to run the program - both interactively and from the command line. If you interact with it through the windowing system, then Figure 1 shows a screen view of the CmpSpots reporting window.

CmpSpots Report Window

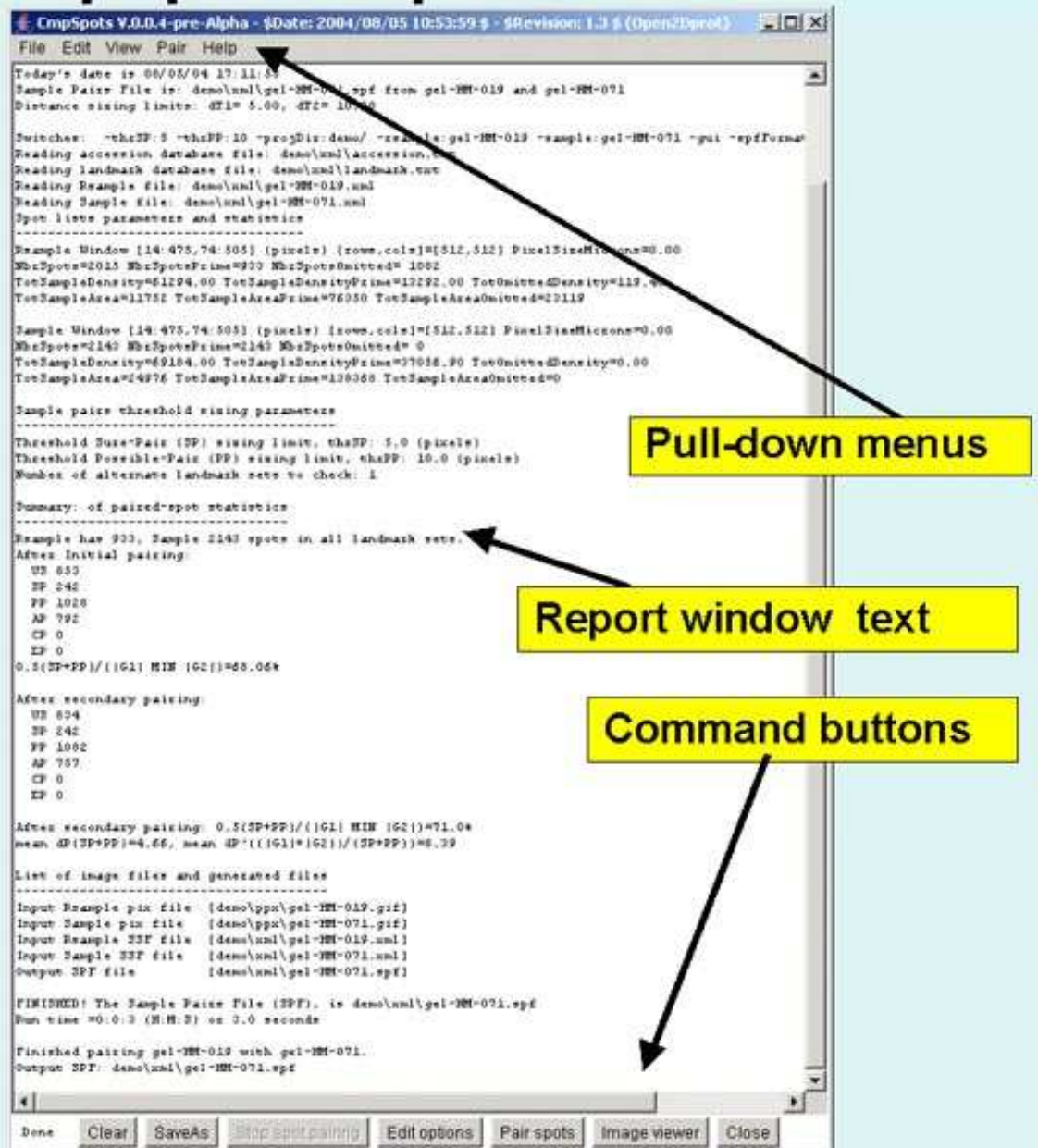


Figure 1. Screen view of the CmpSpots program Report Window interface This screen shot shows the status of CmpSpots in the Report Window graphical user interface after a spot pairing of the demo samples which are 2D PAGE gels. The reference sample or Rsample is gel-HM-019. The other sample denoted Sample is gel-HM-071) has completed. The pairing threshold parameters are shown at the top with the list of **Switches**. The summary statistics on the spot list pairing is summarized in the window including the run time [Run on a 1Ghz Pentium-4 Windows 2000 system]. The Report window has five pull-down menus at the top of the window used to invoke menus: File, Edit, View, Pair, and Help operations. Although, the button commands at the bottom of the window are also available in the menus, they are replicated as buttons for convenience. The **Clear** button will clear the window. **SaveAs** will let you save the Report Window text in a text file. The **Edit options** button pops up a window to let you edit the options including the sample name, computing window, sizing thresholds etc. The **Pair spots** button starts the spot pairing on the current options. You can stop the spot pairing by pressing the **Stop spot pairing** button. After you have paired the spot lists, you can review the paired spot data by pressing the **Image Viewer** to popup the image viewing window. A status area appears in the lower left corner and reports the current state of the spot pairing during an analysis. It shows "Done" since the spot pairing had

completed.

1. Introduction

CmpSpots is an open source Java 2D spot list matching program for finding and pairing spots between the two quantified spot lists. It is part of the Open2Dprot project (<http://open2dprot.sourceforge.net/>). CmpSpots is derived from the GELLAB-II 2D gel spot pairing program **cmpgl2** described in [3]. CmpSpots has been generalized to other types of samples besides 2D gels. While the original program was written in C, CmpSpots is written in Java, uses XML input and output files and has an optional graphical user interface. This initial open-source CmpSpots program code could be used as the basis for more advanced spot pairing methods.

The program may be run either interactively (-gui) or under an OS shell command line interface to implement batch (-nogui). If the default -gui mode is used, after the spot pairing is finished, the user has the option of interactively examining the paired spot data overlaid on the original sample images. The user may also modify the input switch options and save the new options in a "CmpSpots.properties" file in the current project directory when they exit so that the last used options may be used as the default switch options in subsequent running of CmpSpots.

In the analysis of 2D image samples (whether real or virtual), the matching of spots between these samples by pairing spots between their spot lists is difficult to perform manually when there are more than a few spots. Automation is necessary when the samples contain a large number of spots, spots which are at best only locally congruent from sample to sample, when the spots can not be counted on to maintain their "shape" or density and contain little infrastructure on which to build a characterization [3]. There is at best a local congruence between two samples related by some a priori undetermined affine transformation. Some of this description is derived from [3].

Partitioned search - dividing the 2D space into local regions

Aside from the number of spots to be examined, a major problem complicating spot localization may be the global distortions in the samples. However, if we are locally well-behaved data, neighboring spots in one sample will likely be neighbors in another gel but the intervening distances between them will change to a greater degree. The spot pairing problem can be handled various ways.

One method would be to warp (using a non-linear affine transform) one sample image to the geometry of the reference sample image. Then when these images are analyzed by quantifying segmented spots, the spots will be congruent. This method won't work if there are no images (e.g., the 2D spot lists are derived from non-image data (e.g., 2D LC-MS data).

Another method would do a similar warping, but using only the spot lists themselves.

CmpSpots method: A landmark driven spot pairing algorithm

An alternative method is to partition the spot lists into local regions and pair spots within these regions. Within a region, spots surrounding a spot that is known to be paired can most frequently be reliably paired between samples. We will call these spots in the center of the regions "landmarks". The method uses an abstract (x,y) space rather than actual images. Spot (x,y) coordinates are mapped from the sample to the reference sample. This is done, for each landmark, by assigning the landmark spots in the two samples, the cartesian coordinate (0,0). Then using simple linear vector arithmetic, the set of sample spots coordinates are mapped to the reference sample. A set of landmark spots increases the efficiency of inter-sample spot matching by providing an empirical basis for the partitioning of a virtual sample images into tractable corresponding subregions.

A landmark spot may be defined in various ways. In one current empirical procedure for choosing landmarks, it is a morphologically distinctive spot such that neighboring spots and the landmark spot form a consistent morphological structure. Moreover, this structure should be easily recognized across the set of samples used in an experiment consisting of a number of gels (> 2). The landmark spots are selected to cover the regions of interest of the sample fairly evenly if there is uniform distortion between samples. If there are some regions with a lot of distortion, more landmarks should be used in these regions. Depending of the similarity and distortions between the samples, a few to a larger number may be required. This set of landmarks is called the landmark set.

Landmark region

The landmark region is a 2D space surrounding a landmark spot. It is defined as a fuzzy region having more certainty closer in toward the landmark spot. The effective-radius of certainty R_i for landmark i is a distance defined to be half the distance from landmark i to the nearest neighbor landmark. Figure 2 illustrates the effective-radius concept. Images of a spot within the effective-radius of a landmark set would have a higher probability of being aligned (since the landmarks have "perfect" alignment) than if the spot were outside of this radius. Thus, using this heuristic concept of partitioning the spots by landmark region, it is possible to pair the spots automatically once the landmarks are established. The landmark spots are then compared with the two spot lists from the two Sample Spot-list Files (SSF) and the best segmented spot is used rather than the coordinates manually produced. If no spot can be found for a landmark within specified error bounds (currently the maximum latch distance defined using the `-latchLandmarkSpots:maxLatchDist` distance), the landmark coordinate is used in the pairing process. It is possible to check how reliable the particular spot pairing actually was by backchecking paired spots using the Image Viewer.

The probability of finding the same spot in two samples relative to the aligned images of a closeby landmark is greater than if the entire sample spot space were to be searched. This partitioned search has the added advantage that landmark regions contain an order of magnitude fewer spots than the total sample space. Thus the combinatorics of performing the spot matching is greatly decreased as well.

Landmarks could be supplied either using a manual "landmark definition" interactive program (not described here) or might be generated using methods for discovering a subset of robust paired-spots that might be putative landmarks.

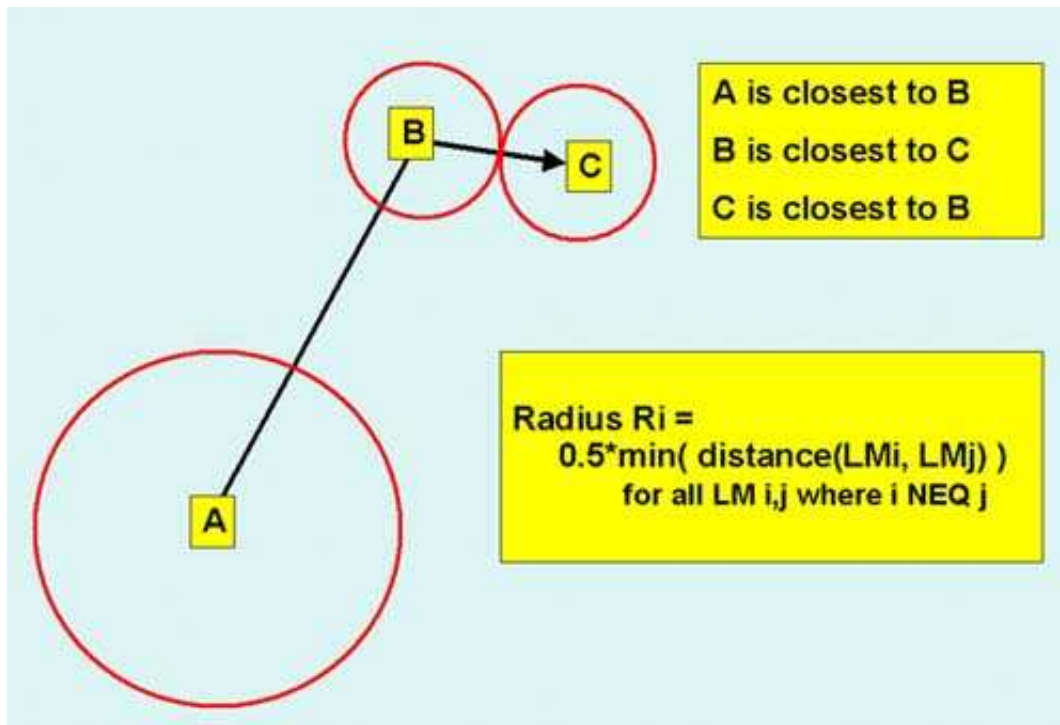


Figure 2. Definition of the effective-radii of a set of landmarks. The effective-radii of certainty of a landmark R_i is $1/2$ the minimum distance from landmark i to its nearest adjacent landmark j . In this example, radius $R_a > R_b$ and $R_b = R_c$. The nearest neighbor of landmark A is B and its next nearest neighbor landmark is C . Estimates of spot pairs within landmark radii are more reliable.

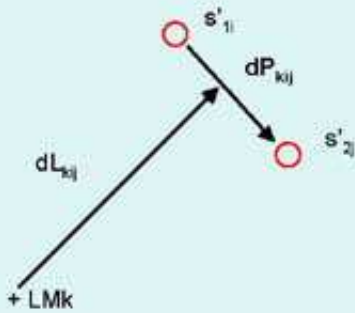
Implementation of landmark-oriented spot pairing between two samples

The spot pairing algorithm is described below. The actual pairing is performed in two passes through the landmark sets data called the primary and gelssecondary pairing procedures. Each procedure operates on one landmark set at a time.

In the primary pairing algorithm (figure 3), the spots are first mapped to the Cartesian coordinate system defined by making the landmark spot $(0,0)$ relative to the origin in the two samples: R_{sample} ($G1$) and $Sample$ ($G2$). Each spot in $G1$ is provisionally paired to the spot that is its nearest neighbor in the projected image of $G2$. Because of possible asymmetry of the two sets the reverse comparison is also performed so that each spot in $G2$ is provisionally paired with its nearest neighbor spot in R_{sample} . The nearest neighbor distance is called d_P (pair distance). The distance d_L is the distance from the landmark spot to the mean locus of the two spots in the provisional pair. Two parameter distances are empirically defined: d_{Tsp} and d_{Tpp} . Spots closer than d_{Tsp} are relatively well paired. Spots greater than d_{T2} are very poorly paired and possibly should not be paired. The default values of d_{Tsp} and d_{Tpp} (5 and 10 pixels respectively) were determined empirically, by examination of the nearest neighbor values of several sets of paired samples under a wide variety of conditions. Figure 4 shows various cases which can occur. Four types of pairing labels can be assigned. There are sure pair "SP", possible pair "PP", ambiguous pair "AP" and unresolved spot "US". The primary spot pair labeling assignments are defined in figure 3.

The distance between paired spots is dP_{kij} and is computed after the two spots (s_{1i}, s_{2i}) are mapped to (s'_{1i}, s'_{2i}) in the same sample space relative to landmark k.

$$dP_{kij} = |s'_{1i} - s'_{2i}|.$$



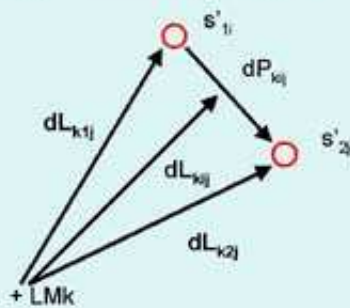
A)

The distance between a spot-pair and the landmark is dL_{kij} and is computed after the two spots (s_{1i}, s_{2i}) are mapped to (s'_{1i}, s'_{2i}) in the same sample space relative to landmark k.

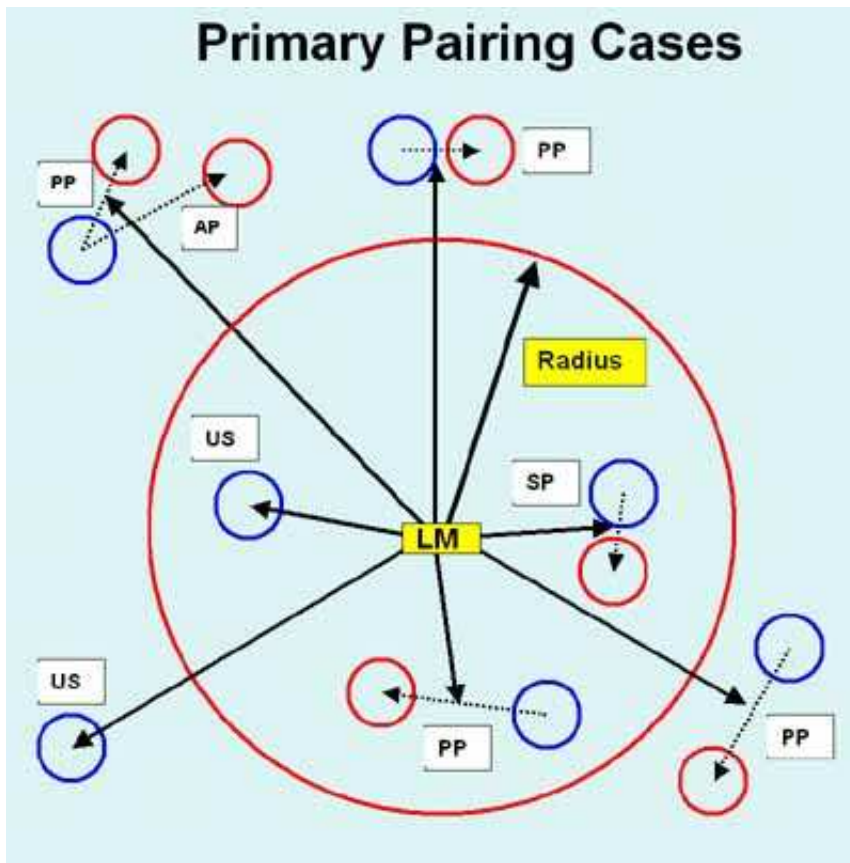
$$dL_{k1i} = \text{distance}(\text{LMk}, s'_{1i}),$$

$$dL_{k2i} = \text{distance}(\text{LMk}, s'_{2i}),$$

$$dL_{kij} = \min(dL_{k1i}, dL_{k2i}).$$



B)



c)

Figure 3. Spot pairing rules. Spot pairing is performed using several features of landmarks and spots in these landmark sets. **A)** Shows the definition of dP or the distance between spots of a spot pair mapped to the same space. **B)** Shows the definition of dL or the distance from a landmark to the spot pair mapped to the same space. **C)** Spot pair primary labeling assignment definitions. Each potential nearest neighbor spot pair in a landmark set has one of four labels: SP - sure pair, PP - possible pair, AP - ambiguous pair, US - unresolved spot. R_k is the effective radius for a landmark set k . The dT_{sp} is the threshold distance for SP spots -**thrSP**: value. The dT_{pp} is the threshold distance for PP spots -**thrPP**: value. primary pairing is described below. The labeling cases are defined by the following cases:

- Case
- [1] US is unresolved spot (no dP) for either sample,
 - [2] SP is $dL_2 \leq R_k$ and $dP_2 \leq dT_{sp}$,
 - [3] PP is $dL_3 \geq R_k$ and $dP_3 \leq dT_{pp}$,
 - [4] PP is $dL_4 \leq R_k$ and $dP_4 \geq dT_{sp}$ and $dP_4 \leq dT_{pp}$,
 - [5] PP is $dL_5 \geq R_k$ and $dP_5 \leq dT_{sp}$,
 - [6] PP is $dL_6 \geq R_k$ and $dP_6 \leq dT_{pp}$. For the other spot
 AP' is $dL_6' \geq R_k$ and $dP_6' \leq dT_{pp}$ and $dP_6' \geq dP_6$.

Secondary spot pairing optimization

A second pass through the data is performed to optimize the spot pairing. Secondary spot pairing can be used to further resolve AP and US labels in adjacent landmark sets into SP or PP labels which are then placed in either of the two sets. There are four cases: (a) two unresolved spots (US and US), (b) two ambiguous pairs (AP and AP), (c-d) one ambiguous spot (AP) and one unresolved spot (US). The new spot pair is upgraded to a SP or PP pairing label and is put into whichever landmark set has the smallest dL for the putative pair. Each landmark set is tested. Each spot has a list of adjacent landmarks, so these are tested to see if pairing can be optimized for that spot. The secondary pairing algorithm is described below.

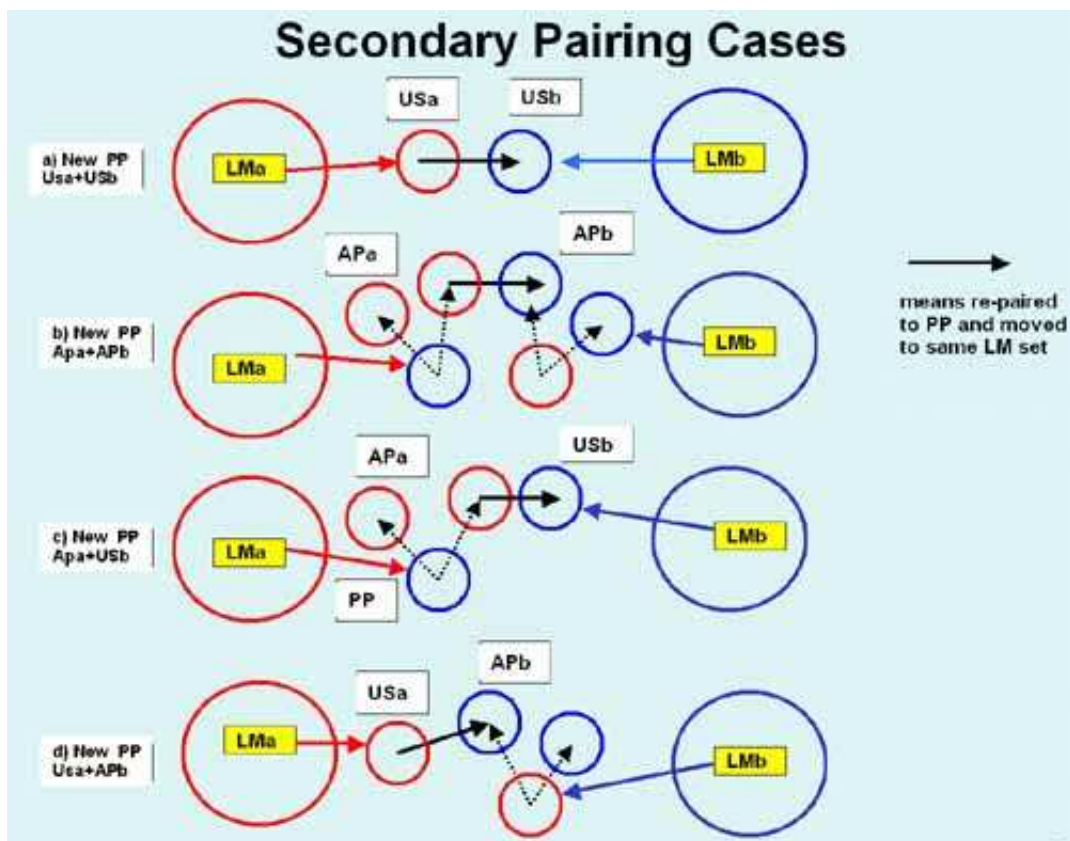


Figure 4. Secondary spot-pairing rules. A second pass through the data is performed to optimize the pairing. Secondary spot pairing can be used to further resolve AP and US labels in adjacent landmark sets into SP or PP labels which are then placed in either of the two sets. There are four cases: (a) two unresolved spots (US and US), (b) two ambiguous pairs (AP and AP), (c-d) one ambiguous spot (AP) and one unresolved spot (US). The new spot pair is upgraded to a SP or PP pairing label and is put into whichever landmark set has the smallest dL for the putative pair. Each landmark set is tested. Each spot has a list of adjacent landmarks, so these are tested to see if pairing can be optimized for that spot.

Project directory structure for Open2Dprot and CmpSpots

All Open2Dprot programs assume a project directory structure. This must exist for the program to proceed. You can either create the structure prior to running any of the programs or you can create it on the fly using the `-projDir:user-project-directory`. It will lookup and/or create the following sub-directories inside of the `user-project-directory`.

```

batch/ directory holding temporary batch files - [NOT USED by CmpSpots]
cache/ directory holding temporary CSD cache files - [NOT USED by CmpSpots]
ppx/ directory holding your original gel input files
rdbms/ directory holding CSD database RDBMS files - [NOT USED by CmpSpots]
tmp/ directory holding generated sample image files
xml/ directory that holds accession DB, landmark DB,
      SSF spot-list files, and generated SPF paired spot list file
      generated by CmpSpots

```

The use of these directories is discussed in the rest of this document.

Input spot list files

The spot lists to be paired are specified by their sample names (e.g., plasma27). The sample and reference samples to be paired is specified by its image file name using the `-sample` and `-rsample` switches with or without the file extension (e.g., `-sample:plasma27.tif` or `-sample:plasma27`). The file extension is determined by looking up the image in the `ppx/` project subdirectory at run time.

The application looks up the sample in the accession database (in `xml/accession.xml` or as specified using the `-accessionFile` switch) and gets additional information about the sample. The Open2Dprot <http://open2dprot.sourceforge.net/Accession> pipeline module is used for entering samples into the accession database.

[Status: The Open2Dprot Accession module program is not released yet. The accession database could be edited manually as either XML (`accession.xml`), or tab-delimited text (`accession.txt`) with Excel.]

The samples correspond to Sample Spot-list Files (SSF) and should be found in the *user-project-directory/xml/* sub-directory. The SSF format is described in the Open2Dprot Seg2Dgel program. SSF files may be used by CmpSpots as either XML or full tab-delimited data formats.

There may be images associated with the samples that can be used with the Image Viewer. This is the case if the spots were derived from real images (e.g., 2D gels) or virtual images (e.g., from synthesized 2D LC-MS data), etc. These images may be in TIFF (.tif, .tiff), JPEG (.jpg), GIF (.gif), or PPX (.ppx GELLAB-II) format. TIFF images may be 8-bits/pixel through 16-bits/pixel, whereas JPEG, GIF, and PPX are 8-bit images. Gray values in the image files have black as 0. This is mapped to 0 for white and the maximum pixel value for black.

The input sample image files, if any, are kept in the *user-project-directory/ppx/* sub-directory. This database directory structure is consistent with and is used by the other Open2Dprot analysis pipeline programs.

Landmark database

A small (5 to 25 depending on the rubber-sheet distortion between samples) set of corresponding landmark spots is required for the Sample and Rsample. These are defined in a landmark database (`xml/landmark.xml`) that uses the same sample names as used in the accession database. The Open2Dprot <http://open2dprot.sourceforge.net/Landmark> pipeline module is used for defining landmark pairs for (Rsample, Sample) into the landmark database.

Output paired-spot list file - the Sample Pairs File (SPF)

The CmpSpots output is a quantified paired-spot list in various ASCII formats including XML and tab-delimited as well as the historical GELLAB-II SPF formats.

The data output file is called the Sample Pairs File (SPF) and is saved in the *user-project-directory/xml/* directory. The generated name the same as the base name of the image file but with a different extension depending on the output format. The possible extensions are: `.spf` (for ASCII format compatible with GELLAB-II), `.xml` (XML format), and `.txt` (tab-delimited format). One of these formats are specified by the `-spfFormat:{F | G | T | X}` command line switch.

[STATUS: `-spfFormat:X` is the default. Note that the XML generated will change with changes in MIAPE.]

1.1 Investigating spot pairing using the paired-spot Image Viewer

Spot pairing may be investigated using the **Image Viewer** button which pops up the Image Viewer window. This lets you select spots to review the pairing data including the quantified spot data. You may add overlays for (subsets of) pairing labels, pairing vectors, landmarks names, landmarks radii. The current selected spot's landmark may be used to filter the spots belonging to that landmark set. There is also a pairing feature histogram that may be used for spot filtering as well. There are four Image Viewer pull-down menus: UL>

- File menu
- View menu
- Filter menu
- Histogram menu

1.1.1 Image viewer - File menu

These commands are used to change the sample image being displayed.

- **Sample Images** submenu - specify the reference sample.
 - **Rsample** - display the Rsample image for overlays
 - **Sample** - display the Sample image for overlays
 - **LM Rsample** - display the Rsample image for viewing the landmark half radii circles
 - **LM Sample** - display the Sample image for viewing the landmark half radii circles
- Checkbox on **Log text to Report Window** - copy all text output from the Image View to the Report window.
- **Save image** - save image to a GIF file.

- **Quit** - exit the program when in GUI mode.

1.1.2 Image viewer - View menu

These commands are used to change the sample image overlays being displayed.

- Checkbox on **Add image overlay, else white** - add overlay
- Checkbox off **Add pairing vectors overlay** - add overlay
- Checkbox on **Add pairing labels overlay** - add overlay
- Checkbox on **Add landmarks overlay** - add overlay
- Checkbox on **Add landmark radii overlay** - add overlay

- Checkbox on **User mouse drag to select spots** - select spots by dragging the mouse over spots rather than clicking on them.

- Checkbox on **Report all spot pairs for selected spot** - report in the text area all paired spots incling all AP labeled spot pairs for the selected spot.

1.1.3 Image viewer - Filter menu

These commands are used to change the histogram filter.

- Checkbox on **Filter spots by histogram bin** - to enable histogram filtering
- Checkbox on **Test spot data GEQ histogram data** - otherwise use data less than the histogram bin data you selected.

- **Filter spots by pairing label**Submenu - select histogram filter to use
 - Checkbox on **Add spots with Sure-Pair (SP) label** - filter by SP labels
 - Checkbox on **Add spots with Possible-Pair (PP) label** - filter by PP labels
 - Checkbox on **Add spots with Unresolved-Spot (US) label** - filter by US labels
 - Checkbox on **Add spots with Ambiguous-Pair (AP) label** - filter by AP labels
-
- Checkbox on **Filter spots in current landmark set** - else use all spots in all landmark sets

1.1.4 Image viewer - Histogram menu

These commands are used to change the histogram filtering criteria.

- **Histogram data**Submenu - how to analyze the histogram data
 - Radio button on **Dist between spots in a pair** - filter by dP or the distance between the two spots after mapped to the same landmark
 - Radio button on **Dist between pair and LM** - filter by dL or distance between the center of the spot pair and their landmark
 - Radio button on **Labels SP,PP,US,AP** - filter by pairing labels labels
 - Radio button on **Landmark set size** - filter landmark set size

2. CmpSpots spot pairing algorithm

The spot pairing algorithm is described in two parts. The first part describes the global processing operations. The second part describes individual spot pairing operations.

Global processing algorithm

1. The program first loads a sample accession file that contains the names of all samples in the database.
2. The program then loads the two sample SSF spot-list files after verifying them with the accession database. One for the Reference sample and one for the sample being paired. We refer to these as "Rsample" and "Sample".
3. It then loads the set of landmarks for the (Rsample,Sample) pair of samples.
4. The landmark effective radii are computed by finding the distance from each landmark to its closest landmark (for both the Rsample and the Sample). Then a landmark's radius is estimated as 1/2 the minimum of these minimum distances for that landmark (for Rsample and Sample). [Spot pairs inside of this radius can become SP pairs, otherwise they can only become at best PP pairs. This is not counting AP pairs which can occur both inside and outside the radius.]
5. The spots for each sample are assigned to their closest landmark and the spots associated with each landmark are called its landmark set. Each spot also tracks a list of next-nearest landmarks used in secondary pairing.
6. For each landmark, spots in its associated landmark set are optimally paired with spots in its corresponding landmark set in the other sample. Mutually-paired spots are assigned Sure-Pair (SP) and Possible-Pair (PP) labels if distances (described below) are within threshold ranges; The case of N:1 or 1:N paired spots between samples are assigned Ambiguous-Pair (AP) labels (i.e.,