

Application of Sequence Alignment Algorithms in Location Tracking Data to Determine Patient State in a Clinical Process

Mark Meyer, MD MPH

Laboratory of Computer Science
Massachusetts General Hospital

Harvard-MIT Division of Health Sciences and Technology
HST951 Final Project

Abstract:

Location tracking systems are becoming more widespread in healthcare, but novel applications using the wealth of data produced are still sparse. Utilizing a sequence alignment algorithm, we have demonstrated their application in healthcare with location tracking data to determine the type of patient based on predefined templates and the current stage of a patient in a clinical process. Future work will further refine the sequence alignment algorithm for use with location tracking data and clinical systems.

Introduction

While retail industries and defense establishment, represented by Wal-Mart and the Department of Defense,^{1,2} have embraced RFID and location tracking systems, their application in the healthcare setting is still in its infancy. However, such systems hold great promise to streamline patient care, manage assets and provide a safer hospital environment. Massachusetts General Hospital and Brigham & Women's Hospital have installed the Radianse location tracking system, a system consisting of a network of receivers and tags that utilize both active RFID and IR to determine location. The use of real-time location data has already been demonstrated at Massachusetts General Hospital to aid in correcting "wrong patient-wrong location" errors in the operating room³ and the concept of real-time patient routing systems has been proven using barcode scanning to track patients through the perioperative process and in the emergency department.^{4,5} The use of an RFID-based location tracking system provides a more robust, sophisticated and automated mechanism to track patient progress through a clinical workflow and to isolate and identify areas in the hospital workflow where efficiency and patient satisfaction may be improved.

The data stream produced by a location tracking system may be represented by a sequence of receivers consisting of the path taken by an individual tag. As the tag and the item or individual to which the tag is attached moves from point to point, a continuous data stream is produced consisting of the receivers encountered between these two points. However, there are issues with location tracking systems leading to a low signal-to-noise ratio and a challenge in the correct interpretation of such data. The Radianse system in particular has a ten second temporal resolution thus if one is able to reach, traverse, and exit a receiver's effective radius, then that receiver will not appear in the tag's history of travel. In addition, the system may "lose" and later "find" a tag that has not actually changed location. One very frequent reason for such a finding is that the patient rolled onto the tag or otherwise is obstructing its function. Under these circumstances, the IR component does not help maintain a connection with the system and the result is that the tag will occasionally drop off the system or the tag will jump between spatially related receivers.

To address issues of information integrity and quality, a mechanism is needed to address the large amount of noise present in the data stream. We hypothesize that utilizing sequence alignment algorithms, originally used in molecular biology, will help compensate for the location data stream's inherent fuzziness and may be used to infer both type of patient based on the sequence of locations they visit and the current state in an already mapped clinical process.

Methods

Sequence Alignment:

The sequence alignment algorithm was written utilizing PHP scripting. We utilize the alignment algorithm where we let:

- $x = x_1 \dots x_n$ and $y = y_1 \dots y_m$ strings over Σ
- special symbol $-$ which is not a member of Σ

Then an alignment of x and y is:

- a pair of strings x' and y' , $|x'| = |y'| = k$, $\max(n, m) \leq k \leq m + n$ over the union of Σ and $\{-\}$ such that:
 - at no position can the strings x' and y' both contain the special symbol $-$, and
 - by removal of all occurrences of $-$ in both x' and y' we get x and y , respectively both hold.

We additionally define the function d which represents the "distance" between the elements of the union of Σ and $\{-\}$ such that for a in Σ :

$$d(a,-) = d(-,a) = g(a)$$

for some suitable function g .

Therefore, $d(a,b)$ is the cost of mutating a to b and $g(a)$ is the cost of inserting or deleting the letter a.

We begin by constructing a matrix representing all possible moves to create all possible sequence alignments. We arbitrarily define the data stream to be x and the template to be y . We also have defined

$g=1$, $d(a, a)=0$ for all a , and $d(a, b) = 1$ for all distinct a and b . We define $D_{0,0}=0$. For each move, we select the minimum of the three possible moves into the square. If we are trying to allocate $D_{i,j}$ then we select the $\min\{D_{i-1,j} + d(x_i,-), D_{i-1,j-1} + d(x_i, y_i), D_{i,j-1} + d(-,y_i)\}=D_{i,j}$.

To reconstruct the optimal sequence, we have that:

- the cost of the optimal alignment $D(x, y)$ is $D_{n,m}$
- at position i, j , a step
 - right represents appending – to x'
 - down represents appending – to y'
 - down and to the right represents x_i and y_j to x' and y' respectively

We use these rules, starting at $D_{n,m}$, to reconstruct the optimal sequence. At any point $D_{i,j}$, we select the next move to be $\min\{D_{i-1,j}, D_{i-1,j-1}, D_{i,j-1}\}$ with a bias for $D_{i-1,j-1}$ in case of a tie until we reach $D_{0,0}$. Using these moves and the above rules, the optimal sequence is reconstructed.

Location Data:

The location tracking data was cleaned utilizing a PHP script. The location tracking data originally is obtained as a MySQL database and through a custom query, all events for a tag may be extracted. This raw data is further processed to eliminate duplicate entries and to eliminate uninformative events. There are a total of twenty-three events in the Radianse system. Events retained during the data cleaning process include:

- assign tag – occurs when a tag is first assigned to a patient or item
- new location – occurs when the tag is found by the system
- change location – occurs when a tag moves from one receiver to another
- unassign tag – occurs when the tag is unassigned from a patients or item
- button0 – occurs when one of the buttons is pushed while within range of a receiver
- button1 - occurs when one of the buttons is pushed while within range of a receiver
- button0_outofrange - occurs when a tag comes back onto the system and one of the buttons is pushed while out of range of a receiver
- button1_outofrange - occurs when a tag comes back onto the system and one of the buttons is pushed while out of range of a receiver

Workflow Template:

We have selected needle localization for breast biopsy patients as the test subjects in this study. This group of patients is ideally suited for study in this application due to their clinical process. During the course of a patient's clinical workflow in needle localization, they spend time both in surgery and radiology. As such, they visit the same clinical location multiple times as they progress through different stages in their care. Therefore, location does not paint a complete picture as the same location may represent very different clinical stages.

We have created a template for needle localization patients represented as:

‘chk’, ‘wt’, ‘chng’, ‘sdsur’, ‘hall’, ‘exam’, ‘hall’, ‘trha’, ‘tr’, ‘sdsur’

where:

- chk – check-in area
- wt – waiting room
- chng – changing room
- sdsur – same day surgery unit
- hall – hall between same day surgery and the radiology suite
- exam- exam room in the radiology suite
- trha – transient recovery holding area
- tr – operating room

There may be many variations in this template as patients may visit, for instance, the same day surgery or transient recovery holding areas several times in their clinical process.

Results

The initial cleaned location tracking data that we used is as follows:

ACC Check-In	Button1	2005-12-07 06:54:31
ACC Atrium Waiting	Change Location	2005-12-07 07:12:06
SDSU Changing Area	Change Location	2005-12-07 07:17:26
ACC Check-In	Change Location	2005-12-07 07:19:24
Tea & Toast	New Location	2005-12-07 07:20:23
TRHA 6-9	Change Location	2005-12-07 07:26:28
Hall Outside TRHA	Change Location	2005-12-07 07:28:24
SD Hall by RN Station	Change Location	2005-12-07 07:28:47
TRHA 6-9	Change Location	2005-12-07 07:29:29
SDSU Recovery 7-14	New Location	2005-12-07 07:30:38
SDSU Recovery 15-21	Change Location	2005-12-07 07:52:12
SDSU Nurses Station	Change Location	2005-12-07 07:58:13
SDSU Recovery 7-14	Change Location	2005-12-07 07:58:53
SDSU Nurses Station	Change Location	2005-12-07 07:59:28
SDSU Recovery 15-21	Change Location	2005-12-07 09:22:04
Sub Waiting 1 Rm 260	New Location	2005-12-07 09:24:07
Sub Waiting 1 Rm 260	Change Location	2005-12-07 09:24:38
Exam Rms 263A-D	New Location	2005-12-07 09:32:18
Exam Rms 286/288	New Location	2005-12-07 10:51:53
Exam Rms 263A-D	New Location	2005-12-07 10:54:27
Hall btwn White&Wang	New Location	2005-12-07 10:57:17
TRHA 6-9	New Location	2005-12-07 10:58:51
Hall Outside TRHA	Change Location	2005-12-07 11:00:13
TRHA 6-9	Change Location	2005-12-07 11:00:46
Tea & Toast	Change Location	2005-12-07 11:35:49
TR 1	Change Location	2005-12-07 11:36:01
Hall Outside TRHA	Change Location	2005-12-07 13:02:18
SDSU Recovery 7-14	Change Location	2005-12-07 13:03:23
SDSU Nurses Station	Change Location	2005-12-07 14:11:54

This then became the sequence:

'chk', 'wt', 'chng', 'chk', 'tt', 'trha', 'htrha', 'rnst', 'trha', 'sdsur', 'sdsur', 'sdsun', 'sdsur', 'sdsun', 'sdsur', 'rwt', 'rwt',
'exam', 'exam', 'exam', 'hall', 'trha', 'htrha', 'trha', 'tt', 'tr', 'htrha', 'sdsur', 'sdsun'

We used this sequence as our data and the needle localization template:

'chk', 'wt', 'chng', 'sdsur', 'hall', 'exam', 'hall', 'trha', 'tr', 'sdsur'

We obtained the following matrix from the sequence alignment algorithm:

		chk	wt	chng	chk	tt	trha	htrha	rnst	trha	sdsur	sdsur	sdsun	sdsur	sdsun	sdsur	rwt	rwt	exam	exam	exam	hall	trha	htrha	trha	tt	tr	htrha	sdsur	sdsun
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
chk	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
wt	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
chng	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
sdsur	4	3	2	1	1	2	3	4	5	6	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
hall	5	4	3	2	2	2	3	4	5	6	7	7	8	9	10	11	12	13	14	15	16	16	17	18	19	20	21	22	23	24
exam	6	5	4	3	3	3	3	4	5	6	7	8	8	9	10	11	12	13	13	14	15	16	17	18	19	20	21	22	23	24
hall	7	6	5	4	4	4	4	4	5	6	7	8	9	9	10	11	12	13	14	14	15	15	16	17	18	19	20	21	22	23
trha	8	7	6	5	5	5	4	5	5	5	6	7	8	9	10	11	12	13	14	15	15	16	15	16	17	18	19	20	21	22
tr	9	8	7	6	6	6	5	5	6	6	6	7	8	9	10	11	12	13	14	15	16	16	16	16	17	18	18	19	20	21
sdsur	10	9	8	7	7	7	6	6	6	7	6	6	7	8	9	10	11	12	13	14	15	16	17	17	17	18	19	19	20	21

$D_{n,m}$ is 20. This corresponds to the optimal alignment solution:

0	0	0	0	1	2	3	4	5	6	6	7	8	9	10	11	12	13	13	14	15	15	15	16	17	18	18	19	19	20
	chk	wt	chng	chk	tt	trha	htrha	rnst	trha	sdsur	sdsur	sdsun	sdsur	sdsun	sdsur	rwt	rwt	exam	exam	exam	hall	trha	htrha	trha	tt	tr	htrha	sdsur	sdsun
	chk	wt	chng	-	-	-	-	-	sdsur	-	-	-	-	-	-	hall	exam	-	-	hall	trha	-	-	-	tr	-	sdsur	-	-

If we consider an alternate clinical workflow for non-needle localization breast biopsy patients:

'chk', 'wt', 'chng', 'htrha', 'trha', 'htrha', 'tr', 'pacu'

This is similar to a needle localization patient but removes the time in radiology and has the patient leave the OR for the post-anesthesia care unit (PACU). We then obtained the matrix:

		chk	wt	chng	chk	tt	trha	htrha	rnst	trha	sdsur	sdsur	sdsun	sdsur	sdsun	sdsur	rwt	rwt	exam	exam	exam	hall	trha	htrha	trha	tt	tr	htrha	sdsur	sdsun
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
chk	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
wt	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
chng	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
htrha	4	3	2	1	1	2	3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
trha	5	4	3	2	2	2	3	4	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
htrha	6	5	4	3	3	3	3	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	17	18	19	20	21	22	23
tr	7	6	5	4	4	4	4	3	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	18	19	19	20	21	22
pacu	8	7	6	5	5	5	5	4	4	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	19	20	20	21	22

$D_{n,m}$ is 22.

We investigated using the original needle localization template and a truncated data stream:

'chk', 'wt', 'chng', 'chk', 'tt', 'trha', 'htrha', 'rnst', 'trha', 'sdsur', 'sdsur'

to assess the ability of the algorithm to determine the current step in the clinical process. This truncated data stream represents a patient who is yet to leave for the radiology suite which is their next step.

Stepping iteratively through the template using $g(a)=2$ and $d(a,b)=3$, we obtained:

- ('chk') = 20
- ('chk', 'wt') = 18
- ('chk', 'wt', 'chng') = 16
- ('chk', 'wt', 'chng', 'sdsur') = 14
- ('chk', 'wt', 'chng', 'sdsur', 'hall') = 15
- ('chk', 'wt', 'chng', 'sdsur', 'hall', 'exam') = 17
- ('chk', 'wt', 'chng', 'sdsur', 'hall', 'exam', 'hall') = 19
- ('chk', 'wt', 'chng', 'sdsur', 'hall', 'exam', 'hall', 'trha') = 18

- ('chk', 'wt', 'chng', 'sdsur', 'hall', 'exam', 'hall', 'trha', 'tr') = 19
- ('chk', 'wt', 'chng', 'sdsur', 'hall', 'exam', 'hall', 'trha', 'tr', 'sdsur') = 17

The minimum occurred with a template of ('chk', 'wt', 'chng', 'sdsur') and a cost of 14.

Discussion

Utilizing the cleaned location data stream, the resulting matrix had an optimal score of 20 and successfully matched the noisy data stream with the correct stage of the clinical process. In particular, the test patient prematurely visited the transient recovery holding area and the location tracking system did not locate the patient in the hallway to the radiology suite, a sentinel event that the patient is on their way to or from radiology. Despite this, it correctly entered gaps into the template that resulted in an accurate representation of the patient's current state; it matched the transient recovery holding area after the patient was in radiology and made a substitution prior to the radiology exam room to account for the missing hall location.

When the template was changed to represent a different clinical workflow, specifically breast biopsy patients not undergoing needle localization, the resulting optimal score was 22, greater than the 20 found previously when matching to a needle localization template. This would lead one to conclude that this data comes from a needle localization patient or that it is from a non-needle localization breast biopsy patient for whom there was a significant process exception which has caused their workflow to more represent a needle localization patient.

When we break the needle localization template into steps where a template string represents the current step and all previous ones, we can run the sequence alignment algorithm against each of these and presumably the one with the lowest score will be the template demonstrating the current step of the clinical process. After truncating the data stream to represent location data for a patient still in the same day surgery unit whose next step is to go to radiology, we found the minimum optimal score corresponding to a template at the same stage as the data stream. We were therefore successfully able to extract the stage in a clinical process from purely the location data and knowledge of the clinical process in the form of a template.

There are two primary parameters, $g(a)$ and $d(a,b)$, that can be changed to influence the results of the sequence alignment algorithm. For much of the initial work, both of these were set to 1. However, for determining the step in a clinical workflow, it was necessary to increase $d(a,b)$ relative to $g(a)$ such that there would be more penalty for making a substitution. Since clinical steps that are not used in the template would result in a substitution, this causes more of a penalty for these templates that are too long and include steps beyond the current step of the data stream. In this instance, $d(a,b)=3$ and $g(a)=2$ worked well in slightly penalizing a substitution that would incorrectly advance a sequence beyond its correct current step.

In future work, additional refinements will be made to the sequence alignment algorithm along with additional work on the templates and location data processing. In particular, it would be beneficial to map physical receiver locations to generic locations or stages in a clinical process to improve the generalizability of the process. For instance, instead of having 'exam' as a location, specifically the exam room, locations such as this would come under the auspices of the general category 'radiology.' This mapping to more general categories would abstract the templates and more easily allow the process to be applied to other institutions and other location tracking systems. This would also allow the alignment algorithm to take into account repeated locations in a general area to improve its results. For instance, if the individual has moved between several of the radiology receivers, this is better evidence that the patient is actually located there and at that clinical step instead of being an anomalous reading or transient movement through an area.

Additional systems may easily be integrated, allowing for more evidence concerning a patient's stage in a clinical process. Information from radiology systems or perioperative records can help augment decisions concerning a patient's location in a clinical process, helping to provide a complete picture of patient status. However, standing on its own, location data serves adequately. We have shown that sequence alignment

techniques can successfully be used solely with location tracking data to determine the type of patient and the patient's current stage in an established clinical workflow.

Conclusion

We have successfully utilized sequence alignment algorithms to serve a functional role in location tracking data. Specifically, we have demonstrated that sequence alignment algorithms may be utilized to determine the type of patient from which a data stream has come along with determining the location in a clinical process a particular data stream currently is. We have demonstrated this through the implementation of a sequence alignment algorithm using PHP scripting and evaluating sets of test data. Future work will further develop this concept with modifications to the algorithm to address factors of location data sequences not present in nucleotide sequences.

References

1. Navy field hospital tests patient tracking in Iraq. 2003 May 29.
http://www.dcmilitary.com/army/standard/8_11/national_news/23364-1.html. Accessed 2005 Oct 7.
2. Want R. RFID: A Key to Automating Everything. *Scientific American* Jan2004;290(1):56-65.
3. Sandberg WS, Hakkinen M, Egan M, Curran PK, Fairbrother P, Choquette K, Daily B, Sarkka JP, Rattner D: Automatic detection and notification of "wrong patient-wrong location" errors in the operating room. *Surg Innov* 2005; 12: 253-60
4. Rotondi A, Brindis C, Cantees K, DeRiso B, Ilkin H, Palmer J, Gunnerson H, Watkins WD. Benchmarking the Perioperative Process. I. Patient Routing Systems: A Method for Continual Improvement of Patient Flow and Resource Utilization. *Journal of Clinical Anesthesia* 1997;9:159-69.
5. Amber R, Everett V. Emergency Department Patient Tracking: A Cost-Effective System Using Bar Code Technology. *J Emerg Nurs* 1996;22:190-5.

Appendix 1

The PHP script used to extract and clean location tracking data from the Radianse database.

```
<html>
  <head>
    <title>Needle Localization Patient Information</title>
  </head>
  <body>
    <link rel="stylesheet" href="style.css" type="text/css">
    <?php
      $host = "";
      $user = "";
      $password = "";

      mysql_connect($host, $user, $password);
      mysql_select_db("radianse");
      $query = "select * from patienttable where patienttypecode = 3 order by patienttableindex
        desc";
      $needle_pats = mysql_query($query);
      $num_needle_pats = mysql_num_rows($needle_pats);

      //for loop selects out the needle loc patients
      for ($i=0; $i < $num_needle_pats; $i++)
      {
        $needle_pat = mysql_fetch_array($needle_pats);
        echo "<div class=container>";
        echo "<div class=name>".substr($needle_pat["LastName"], 0, 3).",
          ".substr($needle_pat["FirstName"], 0, 2)."</div>";

        $query = "select distinct t1.sessionnumber, t2.description as location, t1.event,
          t3.description, t1.eventtime from lightpakeventtable as t1, lanpaktable as t2,
          systemeventtable as t3, sessiontable as t4, persontable as t5 where
          t1.lanpakindex=t2.lanpakindex and t1.event=t3.systemevent and
          t1.sessionnumber=t4.sessionnumber and t4.personindex=t5.personindex and
          t5.patienttableindex=".$needle_pat["PatientTableIndex"]." and t1.event in
          (2,1,3,16,6,7,21,22)";
        $needle_pats_info = mysql_query($query);
        $num_needle_pats_info = mysql_num_rows($needle_pats_info);

        echo "<table width=100% cellpadding=0>";
        $alternate=0;
        $location=NULL;
        //for loop prints out info for each needle loc patient
        for ($j=0; $j < $num_needle_pats_info; $j++)
        {
          $needle_pat_info = mysql_fetch_array($needle_pats_info);
          if ( $location == $needle_pat_info["location"] &&
            $description == $needle_pat_info["description"] )
            { }
          else
            {
              if ($alternate == 0) {
                echo "<tr class=d0>";
                $alternate++;
              }
              else {
```



```
                echo "<tr class=d1>";
                $alternate--;
            }
            echo "<td
            class=location>".$needle_pat_info["location"]."</td>";
            echo "<td
            class=event>".$needle_pat_info["description"]."</td>";
            echo "<td
            class=time>".$needle_pat_info["eventtime"]."</td></tr>";
        }
        $location = $needle_pat_info["location"];
        $description = $needle_pat_info["description"];
    }

    echo "</table>";

    $location = null;
    $description = null;
    echo "</div><p>";
}
?>
</body>
</html>
```

Appendix 2

The PHP script used for sequence alignment.

```
<html>
  <head>
    <title>Sequence Alignment</title>
  </head>
  <body>
    <link rel="stylesheet" href="style.css" type="text/css">
    <?php

      function display_all($array, $data, $template){
echo "<table border=1>";
      echo "<tr><td></td><td></td>";
      foreach($data as $spotdeux){
        echo "<td align=center>".$spotdeux."</td>";
      }
      echo "</tr>";
      $i=-1;
      foreach($array as $spot){
echo "<tr><td>";
        if ($i > -1) echo $template[$i];
        echo "</td>";
        foreach($spot as $spotdeux){
          echo "<td>".$spotdeux."</td>";
        }
        echo "</tr>";
        $i++;
      }
echo "</table>";
    }

    function find_best($x, $y, $array, $matrix, $data, $template, $new_data,
      $new_template){
      if (($x == 0) && ($y == 0)){
        echo "<br>".$array."<br>".count($array)."<br>";
        $size_best=count($array);
        $array_final = array_reverse($array);
        echo "<table border=1><tr>";
        for ($s = 0; $s < $size_best; $s++){
          echo "<td>".$array_final[$s]."</td>";
        }
        echo "</tr><tr><td></td>";
        $data_final = array_reverse($new_data);
        for ($s = 0; $s < count($new_data); $s++){
          echo "<td>".$data_final[$s]."</td>";
        }
        echo "</tr><tr><td></td>";
        $template_final = array_reverse($new_template);
        for ($s = 0; $s < count($new_template); $s++){
          echo "<td>".$template_final[$s]."</td>";
        }
        echo "</tr></table>";
        return $array;
      }
    }
  }
}
```

```

if (($x != 0) && ($y != 0)) $moves[]=$matrix[$x-1][$y-1];
if ($y != 0) $moves[]=$matrix[$x][$y-1];
if ($x != 0) $moves[]=$matrix[$x-1][$y];
echo "<br>". $x. " ". $y. " ". $array;
$best_move=min($moves);
if ($best_move == $matrix[$x-1][$y-1]){
    $array[] = $matrix[$x-1][$y-1];
    $new_template[]=$template[$x-1];
    $new_data[]=$data[$y-1];
    find_best($x-1, $y-1, $array, $matrix, $data, $template,
        $new_data, $new_template);}
elseif ($best_move == $matrix[$x][$y-1]){
    $array[]=$matrix[$x][$y-1];
    $new_template[]='-';
    $new_data[]=$data[$y-1];
    find_best($x, $y-1, $array, $matrix, $data, $template,
        $new_data, $new_template);}
elseif ($best_move == $matrix[$x-1][$y]){
    $array[]=$matrix[$x-1][$y];
    $new_template[]=$template[$x-1];
    $new_data[]='-';
    find_best($x-1, $y, $array, $matrix, $data, $template,
        $new_data, $new_template);}
else{ echo "Error";}
}
}

$data=array('chk', 'wt', 'chng', 'chk', 'tt', 'trha', 'htrha', 'rnst', 'trha', 'sdsur', 'sdsur',
'sdsun', 'sdsur', 'sdsun', 'sdsur', 'rwt', 'rwt', 'exam', 'exam', 'exam', 'hall', 'trha',
'htrha', 'trha', 'tt', 'tr', 'htrha', 'sdsur', 'sdsun');
$template=array('chk', 'wt', 'chng', 'sdsur', 'hall', 'exam', 'hall', 'trha', 'tr', 'sdsur');

$size_data=count($data);
$size_template=count($template);

$g=1;
$d=1;

$matrix[0][0]=0;

for($x = 1; $x < ($size_data + 1); $x++){
    $matrix[0][$x]=$matrix[0][$x-1] + $g;
    echo "matrix[0][". $x. "] = ".$matrix[0][$x]. "<br>";
}

for($y = 1; $y < ($size_template + 1); $y++){
    $matrix[$y][0]=$matrix[$y-1][0] + $g;
    echo "matrix[". $y. "][0] = ".$matrix[$y][0]. "<br>";
}

for($w = 1; $w < ($size_template + 1); $w++){
    for($v = 1; $v < ($size_data + 1); $v++){
        if($template[$w-1] == $data[$v-1]){
            echo "template[". ($w-1). "] =
            ".$template[$w-1]. " and data[". ($v-

```

```

1)."] = ".$data[$v-1]."<br>";
echo "Compare ".$matrix[$w-
1][$v-1]." " ".$matrix[$w-1][$v] +
$g)." " ".$matrix[$w][$v-1] +
$g)."<br>";
$matrix[$w][$v]=min($matrix[$w-1][$v-1],
($matrix[$w-1][$v] + $g), ($matrix[$w][$v-
1] + $g));
}
else{
echo "template[".$(w-1)."] =
".$template[$w-1]." and data[".$(v-
1)."] = ".$data[$v-1]."<br>";
echo "Compare ".$matrix[$w-
1][$v-1] + $d)." " ".$matrix[$w-
1][$v] + $g)." " ".$matrix[$w][$v-1]
+ $g)."<br>";
$matrix[$w][$v]=min(($matrix[$w-1][$v-1]
+ $d), ($matrix[$w-1][$v] + $g),
($matrix[$w][$v-1] + $g));
}
}
}

display_all($matrix, $data, $template);

$initial=array($matrix[$size_template-1][$size_data-1]);

$best_sequence = find_best($size_template, $size_data, $initial, $matrix, $data,
$template, array(), array());

$size_best=count($best_sequence);
echo $best_sequence."<br>";
echo $size_best."<br>";
for ($s = 0; $s < $size_best; $s++){
echo $s." ";
}

?>
</body>
</html>

```