Matching a Performance to Notation

Sometimes an analysis of music performance requires matching a performance to its corresponding notation, or score. The toolbox provides a function, `dynamicMatch.m` to perform this task. The matcher utilizes dynamic programming techniques, runs in polynomial time and is fully described in (Large, 1993). The algorithm for this task was developed in the context of a study of music production errors (Palmer & van de Sande, 1993). The dynamic programming algorithm finds an optimal match between the two sequences, given a scoring function that is hard coded into the program.

The original program written for (Palmer & van de Sande, 1993) required that the two performances be grouped into chords before matching the sequences. In a recent study of piano performance (Large, Rankin, Fink & Houlton, in preparation) this was found to cause difficulties for very long performances, because often no adequate temporal criterion can be found. In this more recent implementation, the notation matrix is grouped into chords (notes beginning on the same beat are chords); it does not group the performance before the match. It also uses an additional scoring function for timing information, and optimizes the sum of the two scoring functions. Thus, performed notes are matched to notated chords, yielding a many-to-one match. Based on this information, the program goes on to group the performance into chords.

Because of the added complexity of grouping and finding the match simultaneously, the matcher may occasionally match incorrectly (this almost always has to do with identifying chords incorrectly). Thus a graphical match inspector / editor is provided to inspect and correct the output, `gmw.m`.

In the following example, we match one of the performances from (Large, Rankin, Fink & Houlton, in preparation), The Goldberg Variation, Aria, by J. S. Bach. In this example, the performer was instructed to play the piece as written, without any ornaments beyond what was notated in the score. Code for the example is provided in `bachExample.mat`.

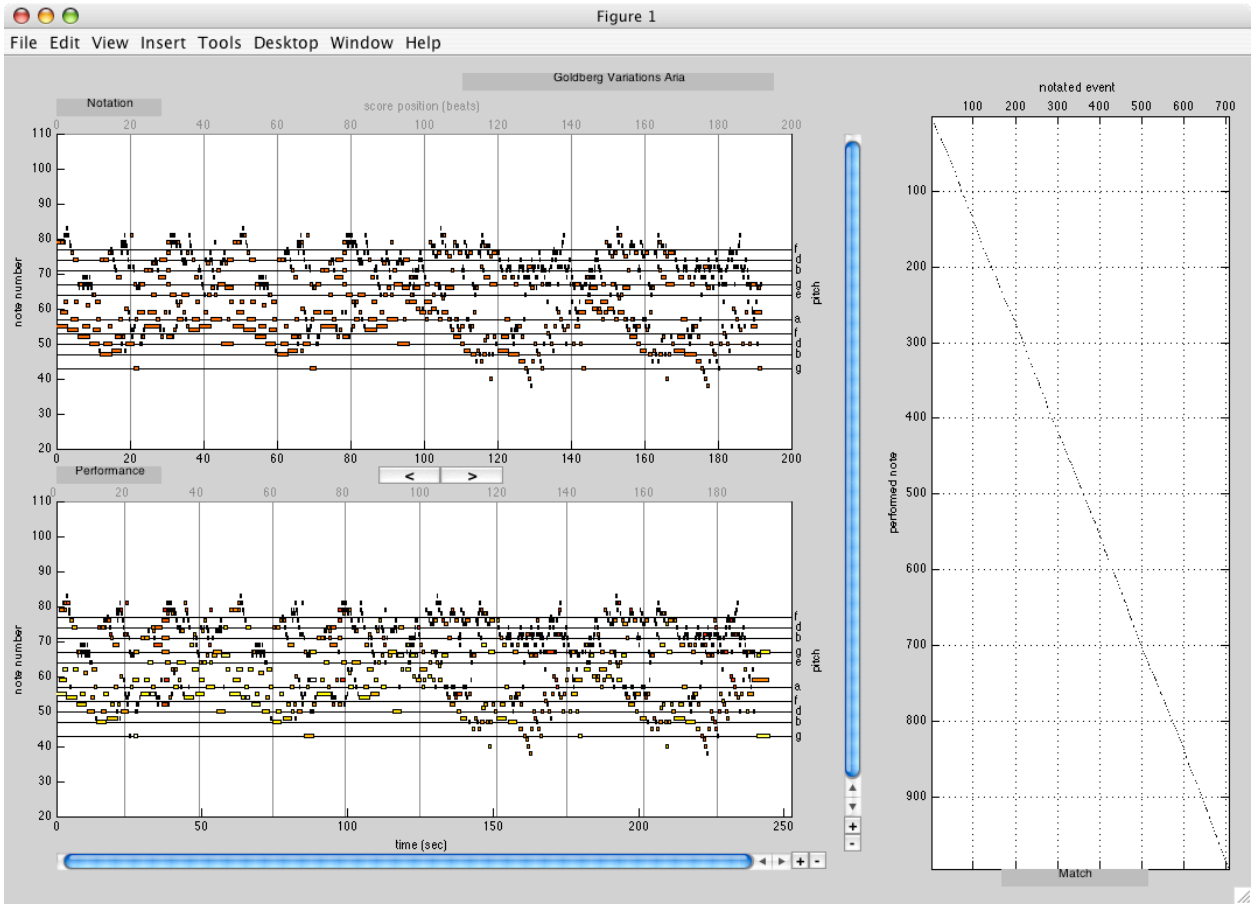
```
>> nmat = readmidi('bachNotation.mid');
>> pmat = readmidi('bachPerformance.mid');

>> M = dynamicmatch(pmat, nmat, 'Goldberg Variations Aria');
>> M

M =

        title: 'Goldberg Variations Aria'
           Nn: [986x7 double]
           Np: [995x7 double]
          GIn: [706x10 double]
          GIp: [995x10 double]
         indN: [1x990 double]
```

```
        indP: [1x990 double]
        mtbl: [995x706 double]
     realGIp: [704x10 double]
          Tn: [704x1 double]
          Bn: [704x1 double]
          Tp: [704x1 double]
          Bp: [704x1 double]
   tempoMap: [704x1 double]
  periodMap: [704x1 double]
   velocMap: [704x1 double]
          Nm: [990x7 double]
           M: {998x11 cell}
```

Now we can inspect the match by typing

```
>> gmw(M)
```

and we see the window



We can also inspect the match by viewing M.M in the Matlab array editor. This produces

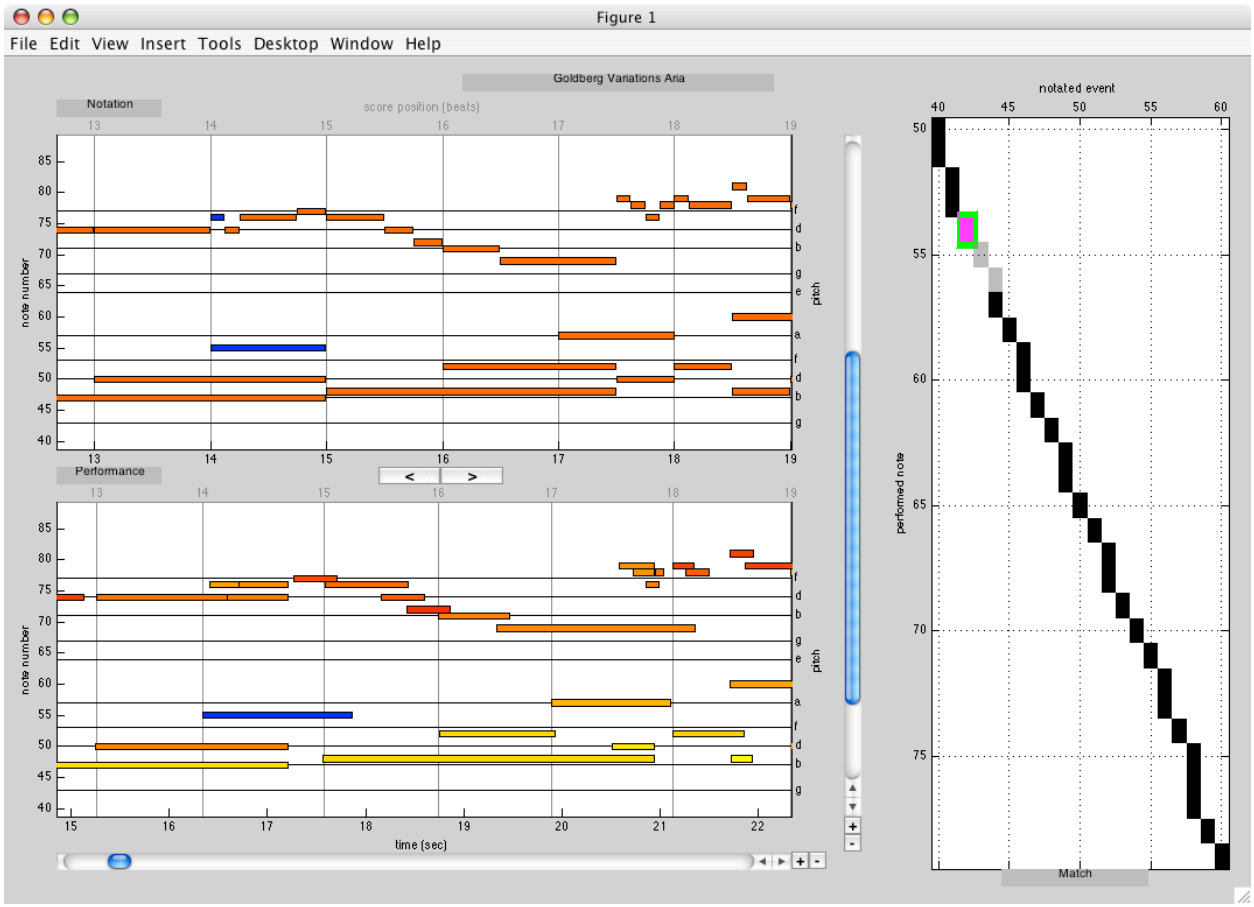
'sub'	1	1	79	0	0	0	64	0	0.9917	0
'M'	1	2	55	55	0	0.0312	64	19	2.9917	3.4479
'M'	2	3	59	59	1	1.1354	64	37	1.9917	2.3438
'M'	2	4	79	79	1	1.151	64	60	0.9917	1.3802
'M'	3	5	81	81	2	2.3542	64	58	0.1167	0.3125
'M'	3	6	62	62	2	2.3646	64	25	0.9917	1.4896
'M'	4	7	79	79	2.125	2.5312	64	59	0.1167	0.9479
'M'	5	8	81	81	2.25	2.6667	64	61	0.4917	0.8125
'M'	6	9	83	83	2.75	3.2604	64	73	0.2417	0.4531
'M'	7	10	54	54	3	3.5729	64	34	2.9917	3.3385
'M'	7	11	81	81	3	3.5781	64	72	0.4917	1.0208
'M'	8	12	79	79	3.5	4.125	64	61	0.2417	0.474
'M'	9	13	78	78	3.75	4.375	64	62	0.2417	0.4115
'M'	10	14	76	76	4	4.6979	64	51	0.4917	0.9427
'M'	10	15	57	57	4	4.7083	64	37	1.9917	0.375
'M'	11	16	74	74	4.5	5.2708	64	48	1.4917	1.6406
'M'	12	17	62	62	5	5.8646	64	33	0.9917	1.474
'M'	13	18	52	52	6	7.0938	64	38	2.9917	3.0052
'M'	13	19	67	67	6	7.1302	64	41	0.1167	0.1875
'M'	14	20	66	66	6.125	7.3125	64	57	0.1167	0.1823
'M'	15	21	67	67	6.25	7.4531	64	54	0.7417	0.7344
'M'	16	22	55	55	7	8.3646	64	36	1.9917	1.4115
'M'	16	23	69	69	7	8.375	64	50	0.1167	0.1719
'M'	17	24	67	67	7.125	8.5625	64	57	0.1167	0.125
'M'	18	25	66	66	7.25	8.6875	64	59	0.1167	0.1198
'M'	19	26	67	67	7.375	8.8021	64	62	0.1167	0.1146
'M'	20	27	69	69	7.5	8.9479	64	62	0.1167	0.2552

Note that the first entry here is note number zero. This is some sort of error with `read-midi`*, because the file plays correctly.*

Now let's carefully examine the match, and make sure that it says what we want. Scrolling down in the array editor, we find:

'M'	40	51	74	74	12	14.1094	64	73	0.9917	1.0208
'M'	41	52	50	50	13	15.25	64	59	1.9917	1.9635
'M'	41	53	74	74	13	15.2552	64	62	0.9917	1.3333
'M'	42	54	55	55	14	16.3438	64	31	0.9917	1.5156
'del'	42	NaN	76	NaN	14	NaN	64	NaN	0.1167	NaN
'sub'	43	55	74	76	14.125	16.4167	64	53	0.1167	0.2969
'add'	44	56	NaN	74	NaN	16.5885	NaN	60	NaN	0.625
'M'	44	57	76	76	14.25	16.7135	64	57	0.4917	0.5
'M'	45	58	77	77	14.75	17.2708	64	70	0.2417	0.4323

Now, we zoom in to that spot using the graphical interface window and see:

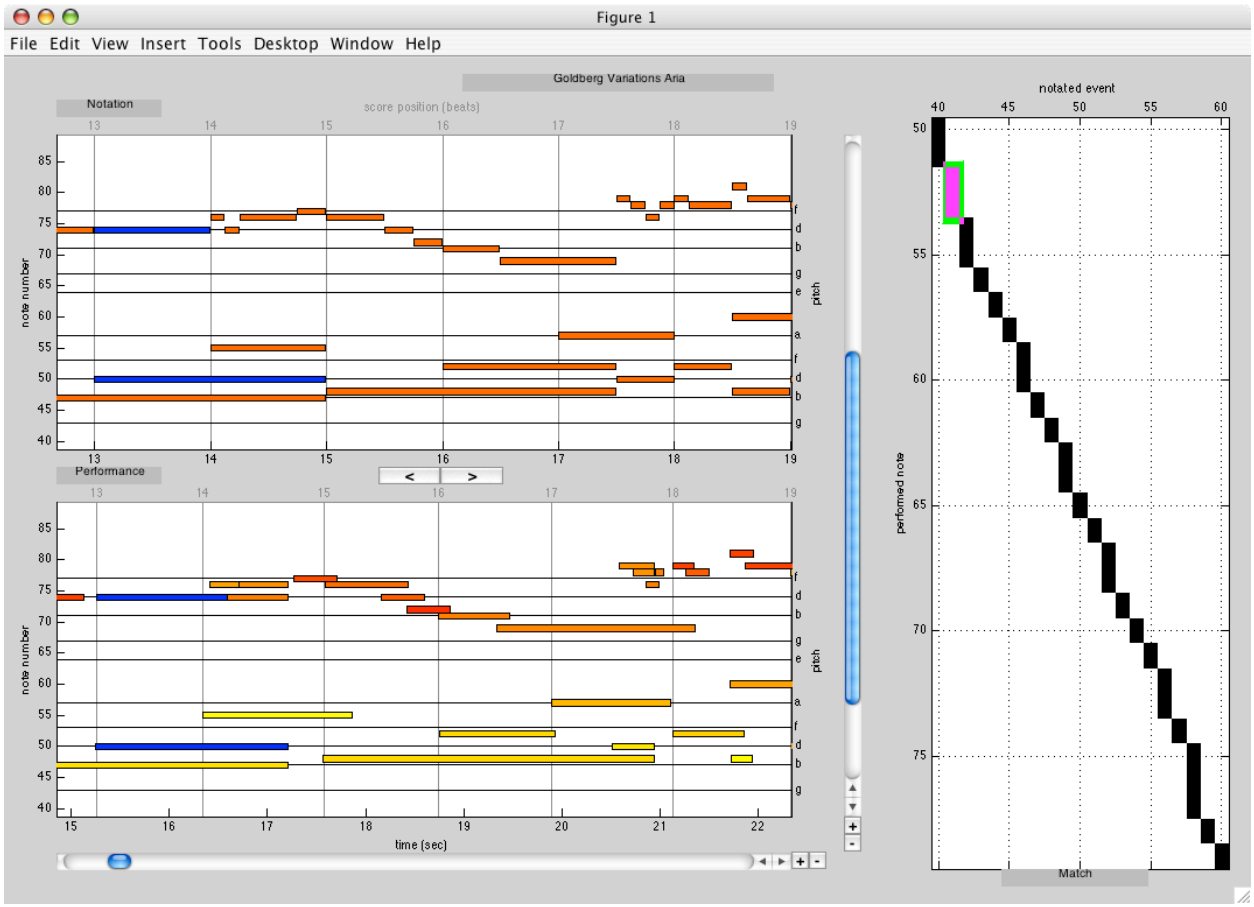


Here the matcher has failed to group note numbers 55 and 76 into a chord, causing some local problems in the match.

To fix this, 1) move the selection back by clicking on the back arrow between the piano roll windows, 2) click on the square 55, 42 and then on the square 56, 43. Then the array editor displays:

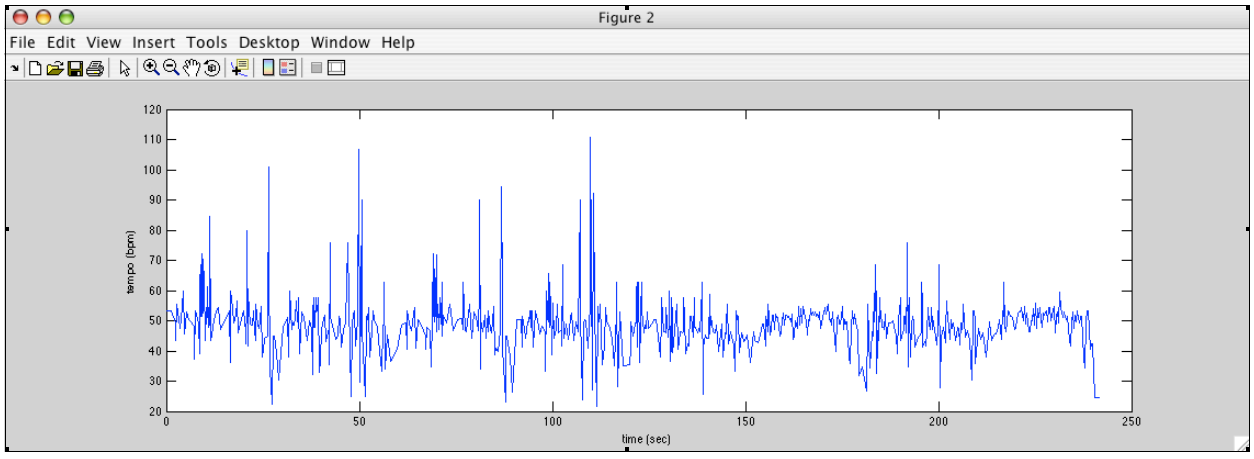
'M'	40	51	74	74	12	14.1094	64	73	0.9917	1.0208
'M'	41	52	50	50	13	15.25	64	59	1.9917	1.9635
'M'	41	53	74	74	13	15.2552	64	62	0.9917	1.3333
'M'	42	54	55	55	14	16.3438	64	31	0.9917	1.5156
'M'	42	55	76	76	14	16.4167	64	53	0.1167	0.2969
'M'	43	56	74	74	14.125	16.5885	64	60	0.1167	0.625
'M'	44	57	76	76	14.25	16.7135	64	57	0.4917	0.5
'M'	45	58	77	77	14.75	17.2708	64	70	0.2417	0.4323

and the match window looks like this:



Note that the array editor updates automatically because we named the match variable M. After each edit, `gmw` automatically writes the variable M into the base workspace.

Take some time, step through and inspect the entire match. Pay particular attention to substitutions, additions and deletions. You will notice the pianist made a few errors, but not many. Also, in this example there is one more correction to be made at location 85, 117. After this edit, the match is correct. Then we can display the tempo map:



References

Large, E. W. (1993). Dynamic programming for the analysis of serial behaviors. *Behavior Research Methods, Instruments, and Computers, 25 (2)*, 238-241.

Large, E. W., Rankin, S. K., Fink, P. (2009). Fractal Tempo Fluctuation and Pulse Prediction. *Music Perception*.

Palmer, C. & van de Sande, C. (1993). Units of knowledge in music performance. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 19*, 457-470.