

CalcPauliTransferEval

```
SetDirectory @ NotebookDirectory[];  
Import["../Link/QuESTlink.m"];
```

Doc

?CalcPauliTransferEval

Symbol

CalcPauliTransferEval[pauliString, ptMaps] returns the full evolution of the given

Pauli string under the given list of PTMap operators. This is often unnecessary to call directly – most users can call ApplyPauliTransferMap[] or DrawPauliTransferEval[] instead – unless you wish to store or process the evaluation history.

The output is a list of sublists, each corresponding to a layer in the evaluation history (i.e. the operation of a PTMap upon the current Pauli string) including the initial Pauli string. Each item therein represents a Pauli product state and has form {prod,id,ancestors} where prod is a Pauli basis state expressed in base-4 digits (see ?GetPauliStringReformatted), id is a unique integer identifying the state, and ancestors is a list of tuples of form {ancId, factor}. These indicate the ancestor Pauli states from which the id'd state was produced under the action of the previous PTMap, and the factor that the map multiplies upon that ancestor state. The basis products of the initial state have ancId=0.

Given the output of CalcPauliTransferEval is stored in variable 'out', it may be easier to interpret the expression resulting from Column @ MapAt[GetPauliString, out, {All,All,1}].

CalcPauliTransferEval accepts the below options:

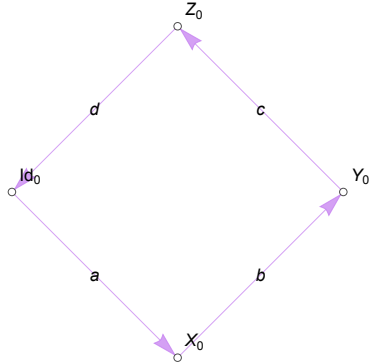
- "CombineStates" -> False which disables combining incident Pauli strings so that the result is an acyclic tree. This means each 'ancestors' list is length-1.
- "CacheMaps" (see ?ApplyPauliTransferMap) which controls the automatic caching of generated PTMaps.
- AssertValidChannels -> False which disables the simplification of symbolic Pauli string coefficients (see ?AssertValidChannels).



Correctness

Maps

```
map = PMap0[0 → {{1, a}}, 1 → {{2, b}}, 2 → {{3, c}}, 3 → {{0, d}}];
DrawPauliTransferMap[map]
```



```
CalcPauliTransferEval[X0, map];
Column @ MapAt[GetPauliString, %, {All, All, 1}]
{{X0, 1, {{0, 1}}}}
{{Y0, 2, {{1, b}}}}
```

```
CalcPauliTransferEval[X0, {map, map, map}];
Column @ MapAt[GetPauliString, %, {All, All, 1}]
{{X0, 1, {{0, 1}}}}
{{Y0, 2, {{1, b}}}}
{{Z0, 3, {{2, c}}}}
{{Id0, 4, {{3, d}}}}
```

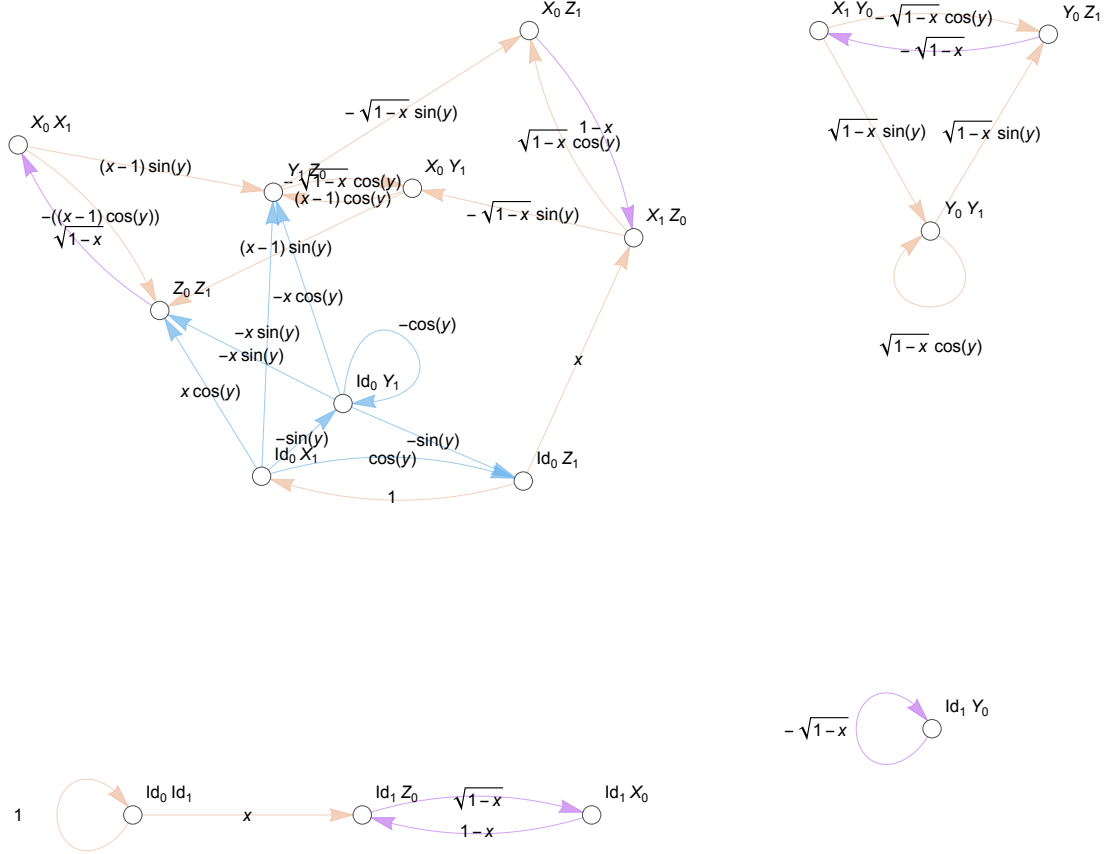
```
CalcPauliTransferEval[X5, {map, map, map, map, map, map}];
Column @ MapAt[GetPauliString, %, {All, All, 1}]
{{X5, 1, {{0, 1}}}}
{{X0 X5, 2, {{1, a}}}}
{{X5 Y0, 3, {{2, b}}}}
{{X5 Z0, 4, {{3, c}}}}
{{X5, 5, {{4, d}}}}
{{X0 X5, 6, {{5, a}}}}
{{X5 Y0, 7, {{6, b}}}}
```

Circuits

```

circ = Circuit[H0 H1 Damp0[x] Rx1[y] ];
DrawPauliTransferMap[circ]

```



```

CalcPauliTransferEval[Y0 Y1, circ]
GetPauliString /@ %[-1, All, 1]]
{
  {{{{2, 2}, 1, {{0, 1}}}}, {{{2, 2}, 2, {{1, -1}}}},
   {{{2, 2}, 3, {{2, -1}}}}, {{{2, 2}, 4, {{3, sqrt(1-x)}}}},
   {{{2, 2}, 5, {{4, Cos[y]}}}}, {{{3, 2}, 6, {{4, Sin[y]}}}}}
}

{Y0 Y1, Y0 Z1}

CalcPauliTransferEval[Y1, circ];
GetPauliString /@ %[-1, All, 1]]

ApplyPauliTransferMap[Y1, circ]
{Y1, Z1, Y1 Z0, Z0 Z1}

-Cos[y] Y1 - x Cos[y] Y1 Z0 - Sin[y] Z1 - x Sin[y] Z0 Z1

```

```

repeated = Join @@ Table[circ, 10];

CalcPauliTransferEval[Y1, repeated];
Plus @@ GetPauliString /@ %[-1, All, 1]

(ApplyPauliTransferMap[Y1, repeated] /. {x → .1, y → .15}) [[All, 2 ;;]]
X1 + X0 X1 + Y1 + X0 Y1 + X1 Z0 + Y1 Z0 + Z1 + X0 Z1 + Z0 Z1
X1 + X0 X1 + Y1 + X0 Y1 + X1 Z0 + Y1 Z0 + Z1 + X0 Z1 + Z0 Z1

```

Options

CombineStates

```

circ = Circuit[H0 H1 Damp0[x] Rx1[y] ];
circ = Join @@ Table[circ, 5];

CalcPauliTransferEval[Y1, circ] [[-1, All, 1]]
Length[%]

CalcPauliTransferEval[Y1, circ, "CombineStates" → False] [[-1, All, 1]]
Length[%]

{{2, 0}, {3, 0}, {2, 3}, {3, 3}, {1, 0}, {1, 3}, {2, 1}, {3, 1}, {1, 1}}
9

{{2, 0}, {3, 0}, {2, 3}, {3, 3}, {1, 0}, {1, 3}, {2, 1}, {3, 1}, {1, 1}, {2, 0},
{3, 0}, {2, 3}, {3, 3}, {2, 1}, {3, 1}, {2, 3}, {3, 3}, {1, 3}, {2, 3}, {3, 3},
{2, 0}, {3, 0}, {2, 3}, {3, 3}, {1, 0}, {1, 3}, {2, 1}, {3, 1}, {1, 1}, {2, 3},
{3, 3}, {1, 3}, {2, 1}, {3, 1}, {1, 1}, {2, 1}, {3, 1}, {2, 1}, {3, 1}, {1, 1},
{2, 0}, {3, 0}, {2, 3}, {3, 3}, {1, 0}, {1, 3}, {2, 1}, {3, 1}, {1, 1}, {2, 0},
{3, 0}, {2, 3}, {3, 3}, {2, 1}, {3, 1}, {2, 3}, {3, 3}, {1, 3}, {2, 3}, {3, 3},
{2, 1}, {3, 1}, {1, 1}, {2, 1}, {3, 1}, {2, 3}, {3, 3}, {1, 3}, {2, 3},
{3, 3}, {2, 3}, {3, 3}, {1, 3}, {2, 3}, {3, 3}, {1, 3}, {2, 3}, {3, 3}}
78

history = CalcPauliTransferEval[Y1, circ];
ancestors = Flatten[history, 1] [[All, 3]];
Length /@ ancestors // Max

history = CalcPauliTransferEval[Y1, circ, "CombineStates" → False];
ancestors = Flatten[history, 1] [[All, 3]];
Length /@ ancestors // Max

2
1

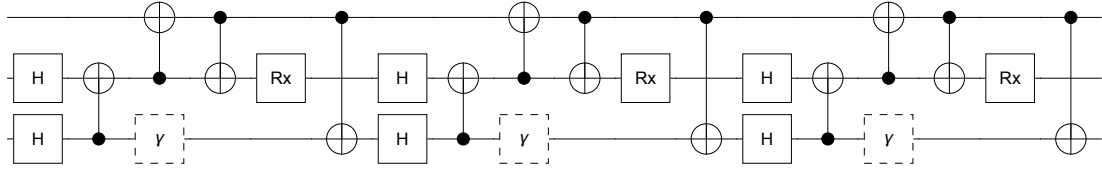
```

CacheMaps

```

circ = Circuit[H0 H1 C0[X1] C1[X2] Damp0[x] C2[X1] Rx1[y] C2[X0] ];
circ = Join @@ Table[circ, 3];
DrawCircuit[circ]

```



```

First @ Timing @ CalcPauliTransferEval[Y1, circ, "CacheMaps" → "Never"]
First @ Timing @ CalcPauliTransferEval[Y1, circ, "CacheMaps" → "UntilCallEnd"]

```

0.091459

0.017185

```

First @ Timing @ CalcPauliTransferEval[Y1, circ, "CacheMaps" → "Forever"]

```

```

First @ Timing @ CalcPauliTransferEval[Y1, circ, "CacheMaps" → "Forever"]

```

0.018647

0.003906

AssertValidChannels

```

CalcPauliTransferEval[Y2 Y1 Z0, Damp0[x]]
CalcPauliTransferEval[Y2 Y1 Z0, Damp0[x], AssertValidChannels → False]
{{{ {2, 2, 3}, 1, {{0, 1}}}}, {{ {2, 2, 3}, 2, {{1, 1 - x}}}}}
{
  {{{ {2, 2, 3}, 1, {{0, 1}}}},
    {
      {
        { {2, 2, 0}, 2, {
          {1,  $\frac{1}{2} (1 - \sqrt{1-x} \text{Conjugate}[\sqrt{1-x}] - \sqrt{x} \text{Conjugate}[\sqrt{x}])$ }}
        },
        {
          {2, 2, 3}, 3, {
            {1,  $\frac{1}{2} (1 + \sqrt{1-x} \text{Conjugate}[\sqrt{1-x}] - \sqrt{x} \text{Conjugate}[\sqrt{x}])$ }}
          }
        }
      }
    }
  }
}

```

Errors

```

CalcPauliTransferEval[Id1, PMap2[{eh}]]

```

... **CalcPauliTransferEval**: Could not pre-compute the Pauli transfer maps due to the below error:

... **CalcPauliTransferMatrix**: Circuit contained an unrecognised or unsupported gate: PMap₀[{eh}]

\$Failed

CalcPauliTransferEval[Id₁, Bop₀]

... **CalcPauliTransferEval**: Could not pre-compute the Pauli transfer maps due to the below error:

... **CalcPauliTransferMatrix**: Circuit contained an unrecognised or unsupported gate: Bop₀

\$Failed

CalcPauliTransferEval[Id₋₁, H₀]

... **CalcPauliTransferEval**: Invalid arguments. See ?CalcPauliTransferEval

\$Failed

CalcPauliTransferEval[1, H₀]

... **CalcPauliTransferEval**: Invalid arguments. See ?CalcPauliTransferEval

\$Failed

CalcPauliTransferEval[X₀, X₀, "CacheMaps" → "x"]

... **CalcPauliTransferEval**: Option "CacheMaps" must be one of "Forever", "UntilCallEnd" or "Never".

\$Failed

CalcPauliTransferEval[X₀, X₀, "CombineStates" → "x"]

... **CalcPauliTransferEval**: Option "CombineStates" must be True or False.

\$Failed

CalcPauliTransferEval[X₀, X₀, "BadOption" → False]

... **OptionValue**: Unknown option BadOption for {CalcPauliTransferMap, ApplyPauliTransferMap}.

\$Failed

CalcPauliTransferEval[]

... **CalcPauliTransferEval**: Invalid arguments. See ?CalcPauliTransferEval

\$Failed