

GetPauliStringReformatted

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

Doc

? GetPauliStringReformatted

Symbol

Reformats symbolic Pauli strings into a variety of other formats convenient for processing.

GetPauliStringReformatted[product, "Index"] returns the integer index of the given Pauli product in the ordered basis of Pauli products. The zero target is treated as least significant.

GetPauliStringReformatted[string, "Index"] returns a list of {index, coefficient} pairs which describe all Pauli products in the given string.

GetPauliStringReformatted[..., "Digits"] returns the individual digits of the basis Pauli string's index (or indices), in base 4, where the rightmost digit is the least significant.

GetPauliStringReformatted[..., "Kronecker"] expands the Pauli string into an explicit Kronecker form. The zero target in the given product corresponds to the rightmost Pauli in the Kronecker form.

GetPauliStringReformatted[..., "String"] returns a compact, string-form of the "Kronecker" format.

GetPauliStringReformatted[..., numQubits] expands the "Digits", "Kronecker" and "String" formats to the specified number of qubits, by padding with '0' digits or 'Id' operators.



? GetPauliString

Symbol

Returns a Pauli string or a weighted sum of symbolic Pauli tensors from a variety of input formats.

GetPauliString[matrix] returns a complex-weighted sum of Pauli tensors

equivalent to the given matrix. If the input matrix is Hermitian, the output can

be passed to Chop[] in order to remove the negligible imaginary components.

GetPauliString[index] returns the basis Pauli string corresponding to the given index, where the returned Pauli operator targeting 0 is informed by the least significant bit(s) of the index.

GetPauliString[digits] specifies the Pauli product via the

base-4 digits of its index, where the rightmost digit is the least significant.

GetPauliString[address] opens or downloads the file at address (a string, of a file location or URL), and

interprets it as a list of coefficients and Pauli codes. Each line of the file is assumed a separate

Pauli tensor with format {coeff code1 code2 ... codeN} (excluding braces) where the codes

are in {0,1,2,3} (indicating a I, X, Y, Z), for an N-qubit Pauli string, and are given in order of

increasing significance (zero qubit left). Each line must have N+1 terms, which includes the

initial real decimal coefficient. For an example, see "https://qtechtheory.org/hamil_6qbLiH.txt".

GetPauliString[..., numQubits] overrides the inferred number of qubits, introducing additional Id

operators upon un-targeted qubits (unless explicitly removed with "Removelds"→False).

GetPauliString[..., {targets}] specifies a list of qubits which the returned Pauli

string should target (in the given order), instead of the default targets {0, 1, 2, ...}.

GetPauliString accepts optional argument "Removelds" → True or False

(default Automatic) which when True, retains otherwise removed Id operators

so that the returned string has an explicit Pauli operator acting upon every qubit.



Correctness

Index

Product

GetPauliStringReformatted[Id₄, "Index"]

0

```

GetPauliStringReformatted[X0, "Index"]
GetPauliStringReformatted[Y0, "Index"]
GetPauliStringReformatted[Z0, "Index"]
1
2
3

GetPauliStringReformatted[X1, "Index"]
GetPauliStringReformatted[Id5 X1 Id0, "Index"]
4
4

GetPauliStringReformatted[Z34, "Index"] == 3 × 434
True

p = Product[Zt, {t, 0, 34}];
GetPauliStringReformatted[p, "Index"] == 434+1 - 1
True

ind = 1245194;
str = GetPauliString[ind]
ind == GetPauliStringReformatted[str, "Index"]
X10 Y0 Y1 Z8
True

```

String

```

GetPauliStringReformatted[X0 + a Y0 + Z3 X0 + a X0 b Y2 c Z4, "Index"]
{{1, 1}, {2, a}, {193, 1}, {801, a b c}}

GetPauliStringReformatted[X0 + a z X0, "Index"]
{{1, 1}, {1, a z}}

GetPauliStringReformatted[Z3 X0, "Index"]
GetPauliStringReformatted[X0 Y2 Z4, "Index"]
GetPauliStringReformatted[X0 + a Y0 + Z3 X0 + a X0 b Y2 c Z4, "Index"]
193
801
{{1, 1}, {2, a}, {193, 1}, {801, a b c}}

GetPauliStringReformatted[X0, "Index"]
GetPauliStringReformatted[1. X0, "Index"]
1
{{1, 1.}}

```

Digits

Product

```
GetPauliStringReformatted[X0, "Digits"]
GetPauliStringReformatted[X0, 5, "Digits"]
GetPauliStringReformatted[X0, "Digits", 10]
{1}
{0, 0, 0, 0, 1}
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1}

GetPauliStringReformatted[Y4, "Digits"]
{2, 0, 0, 0, 0}

GetPauliStringReformatted[X0 Y1 Z2, "Digits"]
{3, 2, 1}
```

String

```
GetPauliStringReformatted[a X0 Y1 Z2, "Digits"]
{{{3, 2, 1}, a}}

GetPauliStringReformatted[a Z5 + b Z5, "Digits"]
{{{3, 0, 0, 0, 0, 0}, a}, {{3, 0, 0, 0, 0, 0}, b}}
```

Kronecker

Product

```
GetPauliStringReformatted[X0, "Kronecker"]
GetPauliStringReformatted[Z0, "Kronecker"]
⊗X
⊗Z

GetPauliStringReformatted[X1, "Kronecker"]
X ⊗ Id

GetPauliStringReformatted[X1, 3, "Kronecker"]
GetPauliStringReformatted[X1, "Kronecker", 4]
Id ⊗ X ⊗ Id
Id ⊗ Id ⊗ X ⊗ Id
```


String

```

str = GetRandomPauliString[4, 5]
GetPauliStringReformatted[str, "String"]
GetPauliStringReformatted[str, "String", 10]
-0.620766 Y1 Y2 Z0 - 0.855372 X2 X3 Y0 Z1 -
  0.120823 Y0 Y3 Z1 + 0.724501 Y0 Z1 Z2 - 0.537396 X1 X2 Y0 Z3

{{IYYZ, -0.620766}, {XXZY, -0.855372},
 {IIZY, -0.120823}, {IZZY, 0.724501}, {ZXXY, -0.537396}}

{{IIIIIIYYZ, -0.620766}, {IIIIIIXXZY, -0.855372},
 {IIIIIIIZY, -0.120823}, {IIIIIIIZZY, 0.724501}, {IIIIIIIZXXY, -0.537396}}

```

Errors

```
GetPauliStringReformatted[Z10, 10, "String"]
```

... **GetPauliStringReformatted**: The given Pauli string targeted a larger index qubit than the number of qubits specified.

\$Failed

```
GetPauliStringReformatted[X-1 Y0 + X2, "Index"]
```

... **GetPauliStringReformatted**: Invalid arguments. See ?GetPauliStringReformatted

\$Failed

```
GetPauliStringReformatted[X-1]
```

... **GetPauliStringReformatted**: Invalid arguments. See ?GetPauliStringReformatted

\$Failed

```
GetPauliStringReformatted[X2, "BadMethod"]
```

... **GetPauliStringReformatted**: Invalid arguments. See ?GetPauliStringReformatted

\$Failed

```
GetPauliStringReformatted[X2, 1, "Kronecker"]
```

... **GetPauliStringReformatted**: The given Pauli string targeted a larger index qubit than the number of qubits specified.

\$Failed

```
GetPauliStringReformatted[X0 X0, "Index"]
```

```
GetPauliStringReformatted[X0 Y0, "Index"]
```

... **GetPauliStringReformatted**: Invalid arguments. See ?GetPauliStringReformatted

\$Failed

... **GetPauliStringReformatted**: Invalid arguments. See ?GetPauliStringReformatted

\$Failed

GetPauliStringReformatted[1]

⋯ **GetPauliStringReformatted**: Invalid arguments. See ?GetPauliStringReformatted

\$Failed

GetPauliStringReformatted[]

⋯ **GetPauliStringReformatted**: Invalid arguments. See ?GetPauliStringReformatted

\$Failed