

NEB Tm API

NEB's public Tm API allows programmatic access under our [Terms of Service](#) to the functionality of the NEB Tm calculator for incorporation in automated high throughput primer design workflows. The API is intended for use in estimating the optimal annealing temperature for PCR with NEB polymerases. Tm values are calculated using thermodynamic data from Santa Lucia [1] and the salt correction of Owczarzy [2]. For Phusion® DNA Polymerases, the salt correction of Schildkraut [2] is used. For additional details, visit the [Help Page](#) of the NEB Tm Calculator.

1. SantaLucia (1998) PNAS 95:1460-5
2. Owczarzy et al (2004) Biochem 43:3537-54

Links to Sections

[Usage](#)
[Tips](#)
[Input Formats and Parameters](#)
[Output Formats](#)
[Sample Code](#)
[Terms of Service](#)

Usage

[Back to top](#)

The API is accessible via GET and POST requests to <https://tmap.neb.com>. For POST requests, the input is expected to be in serialized JSON format including an HTTP 'content-type' header with a value of 'application/json'. Request URI formats are listed below. There are 3 'general purpose' requests and multiple specialized polymerase-specific 'shortcut' requests. The former must be used if specifying a custom monovalent ion concentration instead of a standard NEB polymerase/kit/buffer. A batch option is included as one of the general requests. Output is returned as JSON. The length of the output can be controlled using a 'fmt' parameter (except for batch requests). In addition, there are 2 documentation requests that return product codes and an Open API schema in JSON format, respectively. Detailed tables covering each request category and specifications for each type of output are available below.

API endpoint results are returned as serialized JSON objects. All response objects will contain a 'success' attribute that indicates if the request was completed, a 'data' attribute (possibly empty) that contains the bulk of the response content, and a 'meta' attribute that includes the API version number, a date/timestamp, and the URL used to submit the request. Unsuccessful requests should contain an error attribute in the response indicating the nature of the error. Output formats are shown below. API endpoints will return an HTTP status code of 200 for successfully executed requests and 400 for unsuccessful requests (incomplete or erroneous). Malformed requests may result in a 404 status code as well. Standard 5xx error codes will be returned by the server if the API is non-functional.

Tips

[Back to top](#)

For highest throughput, a POST batch request to /tm/batch is recommended, assuming all calculations are using the same primer concentration and buffer conditions. Sequence IDs are not used at this time, so there are no IDs available in the results; however, the primer sequence is included in the results, and the order of the results is identical to the input order of the sequences in the request.

If a custom buffer condition is needed, use the /tm endpoint, specifying the prodcod as 'custom' and supplying a value for monosalt (mM). In all other cases, monosalt is ignored. The Ta calculated for 'custom' requests is a generic value of 5°C below the lower Tm value.

Notes (e.g. 'Tm difference greater than 5 degrees') are only included in the result when using the /tm endpoint and specifying 'fmt' as 'long'.

In most cases, the second primer sequence (seq2) can be omitted, even for batch requests. The result will only contain a Tm for seq1, and no Ta will be calculated.

Up to 3 ambiguous base codes are allowed per primer sequence. When present, all possible expanded sequences are calculated and min/max Tm values for the set are reported. In these cases, Ta calculations use the min Tm values.

A machine-readable Swagger/OpenAPI v3-compliant specification is available at [/docs/reference](#).

Input Format and Parameters

[Back to top](#)

Parameters

| Name | Description | Type | Allowed |
|----------|---|----------------------|--|
| prodcode | product code representing enzyme/kit/buffer | Text | values available via GET /docs/productcodes |
| conc | primer concentration (mM) | Float | > 0 |
| seq1 | primer 1 sequence | Text | [acgturyswmkbdhvn], min length 8, max length 100 |
| seq2 | primer 2 sequence | Text | [acgturyswmkbdhvn], min length 8, max length 100 |
| seqpairs | array of array of seq pairs for batch request only | [[Text,Text], ...] | [acgturyswmkbdhvn], min length 8, max length 100, seq2 of pair is optional, max pairs per request is 1000 |
| monosalt | equivalent total monovalent Na ⁺ concentration (mM) - used when prodcode is 'custom' - defaults to 50 | Float | > 0 |
| fmt | format of output (defaults to short) | Text | long, short |
| email | contact email address of user | Text | valid email address |

General Requests

[Back to top](#)

| Method | URI | Description | Required Parameters | Optional Parameters |
|---|------------------|-------------------------------------|--------------------------|--|
| POST | /tm | Tm, any product code | prodcode, conc, seq1 | seq2, monosalt (when prodcode is 'custom'), email, fmt |
| Example JSON to post: {"seq1": "AGCGGATAACAATTTACACAGGA","seq2": "GTA AAA CGA CGG CCA GT","conc": 0.5,"prodcode": "q5-0", "email": "tmapi@neb.com", "fmt": "long"} Include HTTP 'content-type' header with a value of 'application/json'. | | | | |
| GET | /tm?<query str> | Tm, any product code | prodcode, conc, seq1 | seq2, monosalt (when prodcode is 'custom'), email, fmt |
| Example: https://tmapi.neb.com/tm?seq1=AGCGGATAACAATTTACACAGGA&seq2=GTA%20AAA%20CGA%20CGG%20CCA%20GT&conc=0.5&prodcode=q5-0&email=tmapi@neb.com | | | | |
| Example: https://tmapi.neb.com/tm?seq1=AGCGGATAACAATTTACACAGGA&seq2=GTAACGACGCCAGT&conc=0.5&prodcode=custom&monosalt=50&email=tmapi@neb.com | | | | |
| POST | /tm/batch + JSON | Tm, batch request, any product code | prodcode, conc, seqpairs | monosalt (when prodcode is 'custom'), email |
| Example JSON to post: {"seqpairs": [{"AGCGGATAACAATTTACACAGGA", "GTAACGACGCCAGT"}], "conc": 0.5, "prodcode": "q5-0", "email": "tmapi@neb.com"} | | | | |

Shortcut Requests

[Back to top](#)

| Method | URI | Polymerase | Required Parameters | Optional Parameters |
|--------|---|-------------------------------|---------------------|---------------------|
| GET | /tm/q5u/{conc}/{seq1}/{seq2}?email={email}&fmt={short long}" | Q5U HS | conc, seq1 | seq2, email, fmt |
| GET | /tm/q5/{conc}/{seq1}/{seq2}?email={email}&fmt={short long}" | Q5/Q5 HS | conc, seq1 | seq2, email, fmt |
| GET | /tm/phusion/{conc}/{seq1}/{seq2}?email={email}&fmt={short long}" | Phusion/Phusion HS Flex | conc, seq1 | seq2, email, fmt |
| GET | /tm/onetaq_std/{conc}/{seq1}/{seq2}?email={email}&fmt={short long}" | OneTaq/OneTaq HS - Std buffer | conc, seq1 | seq2, email, fmt |
| GET | /tm/onetaq_gc/{conc}/{seq1}/{seq2}?email={email}&fmt={short long}" | OneTaq/OneTaq HS - GC | conc, seq1 | seq2, email, fmt |

| Method | URI | Polymerase | Required Parameters | Optional Parameters |
|--------|---|---------------------------|---------------------|---------------------|
| | {email}&fmt={short long}" | buffer | | |
| GET | /tm/taq/{conc}/{seq1}/{seq2}/?email={email}&fmt={short long}" | Taq/HS Taq/EpiMark HS Taq | conc, seq1 | seq2, email, fmt |
| GET | /tm/lataq/{conc}/{seq1}/{seq2}/?email={email}&fmt={short long}" | LongAmp/LongAmp HS Taq | conc, seq1 | seq2, email, fmt |
| GET | /tm/hktaq/{conc}/{seq1}/{seq2}/?email={email}&fmt={short long}" | Hemo KlenTaq | conc, seq1 | seq2, email, fmt |
| GET | /tm/vent/{conc}/{seq1}/{seq2}?email={email}&fmt={short long} | Vent/Deep Vent | conc, seq1 | seq2, email, fmt |

Example: <https://tmap.neb.com/tm/q5/0.5/AGCGGATAACAATTTACACAGGA/GTA%20AAA%20CGA%20CGG%20CCA%20GT?email=tmap@neb.com>

Documentation Requests

[Back to top](#)

| Method | URI | Description | Required Parameters | Optional Parameters |
|--------|-----------------------|----------------------|---------------------|---------------------|
| GET | /tm/docs/productcodes | Valid prodcod values | n/a | n/a |

Example: <https://tmap.neb.com/docs/productcodes>

| | | | | |
|-----|-----------------|--|-----|-----|
| GET | /docs/reference | Open API/Swagger compliant schema (json) | n/a | n/a |
|-----|-----------------|--|-----|-----|

Example: <https://tmap.neb.com/docs/reference>

Output Formats

[Back to top](#)

Output format - JSON - Long Tm

```
{
  "success": BOOLEAN,
  "data": {
    "conc": FLOAT,
    "p1": [
      {
        "tm": INT (melting temp °C),
        "fgc": FLOAT (fractional GC content),
        "len": INT,
        "seq": TEXT
      },
      ...
    ],
    "p2": [
      {
        "tm": INT (melting temp °C),
        "fgc": FLOAT (fractional GC content),
        "len": INT,
        "seq": TEXT
      },
      ...
    ],
    "ta": INT (annealing temp °C),
    "t1min": INT (p1 min melting temp °C),
    "t1max": INT (p1 max melting temp °C),
    "t2min": INT (p2 min melting temp °C),
    "t2max": INT (p2 max melting temp °C),
    "notes": [
      Array of text
    ]
  },
  "meta": {
    "datetime": TEXT,
```

```

        "version": TEXT,
        "url": TEXT
    }
}

```

Output format - JSON - Short Tm

```

{
    "success": BOOLEAN,
    "data": {
        "tm1": INT,
        "tm2": INT,
        "tm1min": INT,
        "tm2min": INT,
        "tm1max": INT,
        "tm2max": INT,
        "ta": INT
    },
    "meta": {
        "datetime": TEXT (ISO-8601 format),
        "version": TEXT,
        "url": TEXT
    }
}

```

Output format - JSON - Batch Tm

```

{
    "success": BOOLEAN,
    "data": [
        {
            "seq1": TEXT,
            "tm1": INT,
            "tm1min": INT,
            "tm1max": INT,
            "seq2": TEXT,
            "tm2": INT,
            "tm2min": INT,
            "tm2max": INT,
            "ta": INT
        },
        ...
    ],
    "meta": {
        "datetime": TEXT,
        "version": TEXT,
        "url": TEXT
    }
}

```

Output format - JSON - productcodes request

```

{
    "success": BOOLEAN,
    "data": [
        {
            "prodcod": TEXT,
            "name": TEXT,
            "catalog": TEXT (catalog #, if applicable)
        },
        ...
    ],
    "meta": {
        "datetime": TEXT,
        "version": TEXT,
        "url": TEXT
    }
}

```

```
}
```

Output format - JSON - Errors

```
{
  "success": BOOLEAN,
  "data": {},
  "error": TEXT,
  "meta": {
    "datetime": TEXT (ISO-8601 format),
    "version": TEXT,
    "url": TEXT
  }
}
```

Sample Code

[Back to top](#)

Sample code - curl

GET example

```
curl -X GET "https://tmapi.neb.com/tm/q5/0.5/AGCGGATAACAATTTACACAGGA/GTAAAAACGACGGCCAGT?email=tmapi@neb.com&fmt=long" -H "accept: application/json"
```

Sample code - Javascript

GET example

```
const request = require('request');

let url = 'https://tmapi.neb.com/tm/q5/0.5/AGCGGATAACAATTTACACAGGA/GTA%20AAA%20CGA%20CGG%20CCA%20GT';

let options = {
  url: url,
  method: 'GET'
};

request(options, (err, res, body) => {
  if (err) { return console.log(err); }
  let json = JSON.parse(body);
  if (json.success) {
    let r = json.data;
    console.log(`Tm1: ${r.tm1} Tm2: ${r.tm2} Ta: ${r.ta}`);
  } else {
    console.log('request failed');
    console.log(json.error[0]);
  }
});
```

POST example

```
const request = require('request');

let url = 'https://tmapi.neb.com/tm/batch';
let seqpairs = [
  ["AGCGGATAACAATTTACACAGGA", "GTAAAAACGACGGCCAGT"],
  ["AGCGGATAACAATTTACACAGGA", "GTAAAAACGACGGCCAGT"]
];

options = {
  url: url,
  method: 'POST',
  body: JSON.stringify({
    'seqpairs': seqpairs,
    'conc': 0.5,
    'prodcode': 'q5-0'
  }),
  headers: {'content-type': 'application/json'}
};

request(options, (err, res, body) => {
  if (err) { return console.log(err); }
  let json = JSON.parse(body);
  if (json.success) {
    let r = json.data;
    for (let i = 0; i < r.length; ++i) {
      console.log(`Seq1: ${r[i].seq1} Tm1: ${r[i].tm1} Seq2: ${r[i].seq2} Tm2: ${r[i].tm2} Ta: ${r[i].ta}`);
    }
  } else {
```

```

        console.log('request failed');
        console.log(json.error[0]);
    }
});

```

Sample code - Python 2.7

GET example

```

import requests
import json

url = 'https://tmapi.neb.com/tm/q5/0.5/AGCGGATAACAATTTACACAGGA/GTA%20AAA%20CGA%20CGG%20CCA%20GT'
res = requests.get(url)

r = json.loads(res.content)

if r['success']:
    print 'Tm1: {} Tm2: {} Ta: {}'.format(r['data']['tm1'], r['data']['tm2'], r['data']['ta'])
else:
    print 'request failed'
    print r['error'][0]

```

POST example

```

import requests
import json

url = 'https://tmapi.neb.com/tm/batch'
seqpairs = [
    ["AGCGGATAACAATTTACACAGGA", "GTAAACGACGGCCAGT"],
    ["AGCGGATAACAATTTACACAGGA", "GTAAACGACGGCCAGT"]
]
input = {
    "seqpairs": seqpairs,
    'conc': 0.5,
    'prodcode': 'q5-0'
}
headers = {'content-type' : 'application/json'}
res = requests.post(url, data=json.dumps(input), headers=headers)

r = json.loads(res.content)

if r['success']:
    for row in r['data']:
        print 'Seq1: {} Tm1: {} Seq2: {} Tm2: {} Ta: {}'.format(row['seq1'], row['tm1'], row['seq2'], row['tm2'], row['ta'])
else:
    print 'request failed'
    print r['error'][0]

```

Terms of Service

[Back to top](#)

Ownership

This API is covered by one or more trademarks and/or copyrights owned or controlled by New England Biolabs, Inc (NEB). All of our rights not expressly granted by the terms of service are hereby retained. You retain your ownership and any associated rights of your own software that may integrate the NEB Tm API service (Service). While NEB develops and validates its products for various applications, the use of a product may require the user to obtain additional third party intellectual property rights for certain applications. For more information, please contact NEB's Global Business Development team at gbd@neb.com.

Access to the API and rules of use

Subject to NEB's the Legal and Disclaimers and the additional restrictions below, you are granted a non-exclusive, revocable, non-sublicensable, limited license to access the API.

By using the API, you agree not to: (A) access the API in ways that circumvent any security measures or probe NEB's systems or test NEB's network vulnerability; or (B) use the API in a manner that in the judgement of NEB constitutes excessive or abusive usage. If you plan on making heavy use of the API, you must first contact tmapi@neb.com for permission.

You may terminate your obligations under the license by discontinuing use of the API. NEB reserves the right to terminate your license without notice to you, and/or to block your access to the API blocked based on your IP address or entire address blocks.

NEB strives to maintain the stability and throughput of the API and provide advance warning of changes or discontinuation; however, the API may be unavailable at times and is subject to change and discontinuation at NEB's discretion.

[Tech Support](#) [Feedback](#) [NEB Overview](#) [Site Map](#)

[Trademarks](#) [Legal and Disclaimers](#) [Privacy](#) [Cookie Policy](#) [Terms of Use](#)

© Copyright 2023 New England Biolabs. All Rights Reserved.